SPATIO-TEMPORAL INTEGRAL EQUATION METHODS WITH APPLICATIONS

Dangxing Chen

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the College of Arts and Sciences.

Chapel Hill
2017

Approved by:

Jingfang Huang

Jianfeng Lu

Jeremy L. Marzuola

Katherine Newhall

Yosuke Kanai

ABSTRACT

Dangxing Chen: Spatio-temporal Integral Equation Methods with Applications
(Under the direction of Jingfang Huang and Jianfeng Lu)

Electromagnetic interactions are vital in many applications including physics, chemistry, material sciences and so on. Thus, a central problem in physical modeling is the electromagnetic analysis of materials. Here, we consider the numerical solution of the Maxwell equation for the evolution of the electromagnetic field given the charges, and the Newton or Schrödinger equation for the evolution of particles. By combining integral equation techniques with new spectral deferred correction (SDC) algorithms in time and hierarchical methods in space, we develop fast solvers for the calculation of electromagnetism with relaxations of the model in different scenarios.

The dissertation consists of two parts, aiming to resolve the challenges in the temporal and spatial direction, respectively. In the first part, we study a new class of time stepping methods for time-dependent differential equations. The core algorithm uses the pseudo-spectral collocation formulation to discretize the Picard type integral equation reformulation, producing a highly accurate and stable representation, which is then solved via the deferred correction technique. By exploiting the mathematical properties of the formulation and the convergence procedure, we develop some new preconditioning techniques from different perspectives that are accurate, robust, and can be much more efficient than existing methods. As is typical of spectral methods, the solution to the discretization is spectral accurate and the time step-size is optimal, though the cost of solving the system can be high. Thus, the solver is particularly suited to problems where very accurate solutions are sought or large time-step is required, e.g., chaotic systems or long-time simulation.

In the second part, we study the hierarchical methods with emphasis on the spatial integral equations. In the first application, we implement a parallel version of the adaptive recursive solver for two-point boundary value problem by Cilk multithreaded runtime system based on the integral equation formulation. In the second application, we apply the hierarchical method to two-layered

media Helmholtz equations in the acoustic and electromagnetic scattering problems. With the method of images and integral representations, the spatially heterogeneous translation operators are derived with rigorous error analysis, and the information is then compressed and spread in a fashion similar to fast multipole methods. The preliminary results suggest that our approach can be faster than existing algorithms with several orders of magnitude.

We demonstrate our solver on a number of examples and discuss various useful extensions. Preliminary results are favorable and show the viability of our techniques for integral equations. Such integral equation methods could well have a broad impact on many areas of computational science and engineering.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisers, Professor Jingfang Huang and Jianfeng Lu, whose guidance and advice has been indispensable over these past four years. I am especially grateful for the freedom that they have allowed in my research, as well as their unique perspective on computational science and mathematical modeling, which will no doubt shape my own philosophy as I move forward.

I would like to thank my collaborators: Dr. Bo Zhang, Dr. Chao Yang, Dr. Wenzhen Qu, Xingjian Guo. Special thanks are given to Professor Lin Lin for his numerous support and the fruitful work and discussion we had together. Without his help, my achievements would not be possible.

I would also like to thank the tremendous support and encouragement I received from other professors at the University of North Carolina, especially from Professor Katherine Newhall and Professor Jeremy Marzuola.

I am indebted as well to my fellow students (past and present), with whom I have shared this incredible journey. Special thanks go to Yan Feng, Yuan Gao, Wenhua Guan, Feng Shi, Hsuan-Wei Lee, Yanni Lai, Fuhui Fang, Tim Wessler, Jason Pearson, Taoran Li, Xianchao Huang, Yifan Yao, Zhengkan Yang, Meiguo Wang. I would also like to express my deep appreciation to the wonderful staff at the University of North Carolina: Laurie Straube.

Last but not least, my girlfriend Jiahui Ye holds all my gratitude for her patience and for the love that she gives to me every day in my life. This thesis is dedicated to her, together with my beloved parents, my mother Zhen Zhou and my father Litong Chen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADI alternative direction implicit. 21

CFL Courant-Friedrichs-Lewy. 11, 21

DAE differential algebraic equation. 24

ETDRK exponential time-diffencing Runge-Kutta method. 54, 55, 58

FFT fast Fourier transform. 61

FMM fast multipole method. 17, 61, 63, 77, 81, 82, 89–91, 93, 94, 98

GMRES generalized minimal residual method. 17

IEM integral equation method. 19, 22, 23, 54, 55, 57–60

IVP initial value problem. 1, 20, 25, 36, 51

KDC Krylov deferred correction. 12, 22, 23, 33, 35, 41, 44, 49, 62

LDA local density approximation. 10

ODE ordinary differential equation. 11, 12, 18, 20, 21, 23, 25, 35–37, 51

PDE partial differential equation. 11–13, 20, 21, 24, 32, 50, 58, 93

RK4 fourth-order Runge-Kutta method. 27

SDC spectral deferred correction. iii, 12, 18, 19, 21–23, 34, 35, 37, 38, 44, 49, 53, 54, 62, 93, 98

TDDFT time-dependent density functional theory. 9–11, 23, 58, 93

TDKS time-dependent Kohn-Sham. 1, 10, 11, 13, 59, 93

CHAPTER 1

Introduction

The study of material in the presence of external electromagnetic field is central to material science and has provided a rich history of mechanistic insights on interesting phenomenon such as electrostrictive, magnetorestrictive, and piezomagnetic effect, etc. Given a collection of particles in the material, the evolution of the electromagnetic field is determined by the Maxwell equation and the motion of particles are determined by Newton or Schrödinger equation. Given its widespread importance, it is clear that accurate electromagnetic analysis is essential for a faithful physical description of properties of a material, and thus for a quantitative understanding of the response of the material in the presence of electromagnetic field at the classical and quantum level.

In this dissertation, we develop new mathematical and computational techniques for the response of the material under applied fields. Specifically, we discuss the applications of electronic structure theory and layered media problems. electronic structure theory is being used today in an ever-increasing range of applications to widely-varying systems in chemistry, biology, solid-state physics, and material science. Among them, the vast majority of electronic structure calculations today lie in spectroscopy and real-time dynamics in non-perturbative fields [1, 2, 3]. In the simulation, our method relies on a continuum model of the nuclei and electrons with external fields and calculate their motions by solving the Ehrenfest molecular dynamics. The time-dependent Kohn-Sham (TDKS) equation in Ehrenfest dynamics is recast as a Picard type integral equation, which gives it a strong mathematical foundation and then solved using pseudo-spectral collocation formulation by deferred correction method with new preconditioning techniques. In contrast to existing temporal solvers, however, our method is capable of stable arbitrary high-order so that the step-size can be optimal. The result is an iterative solver for the initial value problem (IVP) that is accurate, robust, and symplectic.

In our second application, we aim to solve the layered media problem. The problems of designing composite materials that exhibit a specific electromagnetic response is an area of active research

[4, 5, 6]. Examples include the design of random media with a well-defined macroscopic refraction (coherent scattering) [5] and the fabrication of metamaterials for cloaking [6], near field imaging and so on. In the calculation, the electromagnetic fields in different layers with external source have to be calculated by solving the Helmholtz equation with complicated boundary conditions. In our method, the Helmholtz equation is reformulated into a second-kind boundary integral equation, which gives it a well-conditioned representation and then solved using a new fast algorithm that can be seen as an example of the hierarchical methods. Different from the existing methods which focus on compressing the integral form of the domain Green's function, however, our algorithm directly compress the free-space Green's function and that the information is translated with modified multipole mapping. The result is a solver for the domain Green's function that is fast, accurate, and robust, with a well-understood mathematical theory and controlled numerical error.

We begin in the next section with a brief introduction of the mathematical formulation of the electromagnetic field, which is a very mature subject [7, 8, 9, 10].

1.1 Electromagnetic field

Consider some moving charged particles in the free space (can be generalized to the case in a matter), what are the fields produced by particles? In the study of electromagnetism in the classical mechanics, the physical laws are governed by the following mathematical equations:

- **Gauss's law:** The net electric flux through any closed surface is proportional to the net electric charge within that closed surface. This observation can be formalized as follows:

$$\oiint_S \vec{E} \cdot \, d\vec{S} = \frac{1}{\epsilon_0} Q,$$

  where $\vec{E}$ is the electric field, Q is the total charge enclosed within the surface, and ϵ_0 is the permitivity of free space. With the help of the divergence theorem and the representation of the charge as integral of the charge density, the equation can be deduced as

$$\iiint_V \nabla \cdot \vec{E} \, dV = \frac{1}{\epsilon_0} \iiint_V \rho \, dV.$$

Since the equation holds for arbitrary closed surface, the differential form can be derived

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0}.$$

- **Gauss's law for magnetism:** The total magnetic flux through a closed surface is zero. Mathematicallly, this implies that

$$\oiint_S \vec{B} \cdot d\vec{S} = 0,$$

where $\vec{B}$ is the magnetic field. With the help of the divergence theorem, it can be deduced that

$$\iiint_V \nabla \cdot \vec{B} \; dV = 0,$$

from which the differential form is derived

$$\nabla \cdot \vec{B} = 0.$$

- **Faraday's Law:** A changing magnetic field induces an electric field: the voltage induced in a closed circuit is proportional to the rate of change of the magnetic flux it encloses. This observation can be formalized as follows:

$$\oint_{\partial\Sigma} \vec{E} \cdot d\vec{l} = -\iint_\Sigma \frac{\partial \vec{B}}{\partial t} \cdot d\vec{\Sigma}.$$

With the help of the Stokes' theorem, it can be derived that

$$\iint_\Sigma \nabla \times \vec{E} \cdot d\vec{\Sigma} = -\iint_\Sigma \frac{\partial \vec{B}}{\partial t} \cdot d\vec{\Sigma}.$$

from which the differential form can be derived

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}.$$

- **Ampére's circuital law:** A changing electric field induces a magnetic field: the magnetic

field induced around a closed loop is proportional to the electric current plus rate of change of electric field it enclosed. The observation can be formalized as the following:

$$\oint_{\partial\Sigma} \vec{B} \cdot d\vec{l} = \iint_{\Sigma} \left(\mu_0 \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} \right) \cdot d\vec{S},$$

where c is the speed of the light and $\vec{J}$ is the electric current. With the help of Stokes' theorem, it can be deduced that

$$\iint_{\Sigma} \nabla \times \vec{B} \; d\vec{S} = \iint_{\Sigma} \left(\mu_0 \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} \right) \cdot d\vec{S}.$$

Then the differential form can be derived as

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t}.$$

Maxwell equation: By the above four laws with mathematical statements, the classical Maxwell equation can be derived as follows:

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0}, \tag{1.1.1}$$

$$\nabla \cdot \vec{B} = 0, \tag{1.1.2}$$

$$\nabla \times \vec{E} - \frac{\partial \vec{B}}{\partial t} = 0, \tag{1.1.3}$$

$$\nabla \times \vec{B} - \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} = \mu_0 \vec{J}. \tag{1.1.4}$$

This is the general form of the Maxwell equation, which summarizes the almost entire theoretical content of classical electrodynamics. In the expression, the fields ($\vec{E}$ and $\vec{B}$) are put on the left whereas the sources (ρ and $\vec{J}$) are put on the right to emphasize that all electromagnetic fields are ultimately attributable to charges and currents. What's more, by applying the divergence to Eq. (1.1.4), the conservation of charge can be derived

$$\nabla \cdot \vec{J} = -\frac{\partial \rho}{\partial t}. \tag{1.1.5}$$

The four equations can actually be reduced to two with the introduction of the potential functions. By doing so, the degree of freedom for the electric field is reduced and the generalization to the quantum mechanics is straightforward. From Eq. (1.1.2), the magnetic field can be rewritten as

$$\vec{B} = \nabla \times \vec{A}. \tag{1.1.6}$$

Then from Eq. (1.1.3), further potential can be introduced such that

$$\vec{E} = -\nabla\phi - \frac{\partial \vec{A}}{\partial t}. \tag{1.1.7}$$

With the help of potentials, Eqs. (1.1.2) and (1.1.3) are satisfield. Plug Eq. (1.1.7) into (1.1.1), it can be found that

$$\Delta\phi + \frac{\partial}{\partial t}\left(\nabla \cdot \vec{A}\right) = -\frac{\rho}{\epsilon_0}.$$

Putting Eqs. (1.1.6) and (1.1.7) into Eq. (1.1.4), after some algebra, the equation is arrived as

$$\left(\nabla^2 \vec{A} - \frac{1}{c^2}\right) - \nabla\left(\nabla \cdot \vec{A} + \frac{1}{c^2}\frac{\partial \phi}{\partial t}\right) = -\frac{1}{c^2\epsilon_0}\vec{J}.$$

To enforce the invariant of gauge transform

$$\vec{A} \to \vec{A} + \nabla\mathcal{X},$$
$$\phi \to \phi - \frac{\partial \mathcal{X}}{\partial t},$$

The Coulomb gauge is imposed so that

$$\nabla \cdot \vec{A} = 0.$$

The advantage of the Coulomb gauge is that the scalar potential is particularly simple to calculate; the disadvantage is that $\vec{A}$ is particularly difficult to calculate. (It is also possible to choose other

gauges such as Lorentz gauge.) Then we end with the coupled equations

$$\Delta\phi = -\frac{\rho}{\epsilon_0}, \tag{1.1.8}$$

$$\left(\nabla^2\vec{A} - \frac{1}{c^2}\right) - \frac{\nabla}{c^2}\frac{\partial\phi}{\partial t} = -\frac{1}{c^2\epsilon_0}\vec{J}. \tag{1.1.9}$$

Under the electromagnetic field, the classical particles feel the Lorentz forces

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}),$$

where $\vec{v}$ is the velocity of the moving charges. Quantum mechanically, particles feels the Hamiltonian operator of the form

$$\mathcal{H} = \frac{1}{2}\left[\vec{\sigma} \cdot \left(-i\nabla - \frac{1}{c}\vec{A}\right)\right]^2 + \phi,$$

where $\vec{\sigma}$ is the Pauli matrices.

To summarize, Maxwell's equations tell us how charges produce fields; reciprocally, the force law or Hamiltonian tells us how fields affect charges. In the following sections, we briefly discuss two applications of electromagnetism.

1.2 Electronic structure theory

The electronic structure theory has many applications and plays an important role in quantum chemistry. Consider some atoms consist of nuclei and electrons in the material. Given the applied external electromagnetic field, what are their motions?

In quantum mechanics, the evolution of the state is governed by the Schrödinger equation:

$$i\frac{\partial\Psi}{\partial t} = \mathcal{H}\Psi,$$

where Ψ is the wave function, $\mathcal{H}$ is the Hamiltonian operator, and atomic units are used. In electronic structure theory, the Hamiltonian with N electrons $\{\vec{r}_i\}$ and N_{nuc} nuclei $\{\vec{R}_I\}$ with mass M_I and

charge Z_I in the presence of external fields can be written as

$$\mathcal{H} = \sum_{I=1}^{N_{nuc}} \frac{P_I^2}{2M_I} + \sum_{i=1}^{N} \frac{p_i^2}{2} + V + V_{ext}.$$

The physical interpretations of the Hamiltonian are the following:

- P_I, p_i are the corresponding momemtum operators of the nuclei and electrons

$$P_I = -i\nabla_I, \quad p_i = -i\nabla_i.$$

- V is the interaction potential between the nuclei and electrons

$$V(\{\vec{R}_I\}, \{\vec{r}_i\}) = \frac{1}{2} \sum_{I \neq J} \frac{Z_I Z_J}{|\vec{R}_I - \vec{R}_J|} + \frac{1}{2} \sum_{i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|} - \sum_{i,I} \frac{Z_I}{|\vec{r}_i - \vec{R}_I|}$$

- V_{ext} is the external potential.

Then the Schrödinger equation in this circumstance reads

$$i\frac{\partial}{\partial t}\Psi(\{\vec{R}_I\}, \{\vec{r}_i\}, t) = \mathcal{H}(\{\vec{R}_I\}, \{\vec{r}_i\}, t)\Psi(\{\vec{R}_I\}, \{\vec{r}_i\}, t).$$

Remark 1.2.1. We want to make some remarks about the model.

- The relativistic effects are neglected in our current formulation. Roughly speaking, the relativistic effects are considered important for heavy atoms, in which the inner electrons are held more tightly to the nucleus and have velocities which approach to the speed of light as the atomic number increase [11, 12].

- For simplicity, the magnetic field is not presented, hence the spin is also neglected. In a diagmagnetic material, all electrons are spin paired and the material does not have a net magnetic field.

This is generally considered the very realistic treatment since it respects the quantum nature of the particles, and is the predominant method used in molecular and electron dynamics simulations. This realism, however, comes at a significant expense as the dimension of the system grows exponentially

with the number of nuclei and electrons. A natural question, therefore, is to ask whether it is possible to treat the nuclei classically. One approach for this is known as the Ehrenfest molecular dynamics [13]. One starts with the separation ansatz for the wave function of the molecular system between the nuclei and electrons [14],

$$\Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) = \Phi(\{\vec{r}_i\},t)\mathcal{X}(\{\vec{R}_I\},t)e^{i\int_0^t \widetilde{E}_e(s)\ ds},$$

where

$$\begin{aligned}\widetilde{E}_e(t) &= \left\langle \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \right\rangle \\ &= \iint \Phi^*(\{\vec{r}_i\},t)\mathcal{X}^*(\{\vec{R}_I\},t)\mathcal{H}_e\Phi(\{\vec{r}_i\},t)\mathcal{X}(\{\vec{R}_I\},t)\ d\vec{r}\ d\vec{R}\end{aligned}$$

and

$$\mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t) = \sum_{i=1}^{N}\frac{p_i^2}{2} + V(\{\vec{R}_I\},\{\vec{r}_i\},t) + +V_{ext}(\{\vec{R}_I\},\{\vec{r}_i\},t).$$

Taking the inner products with respect to Φ and $\mathcal{X}$ and imposing the energy conditions

$$\begin{aligned} i\left\langle \mathcal{X}(\{\vec{R}_I\},t) \mid \frac{\partial\mathcal{X}(\{\vec{R}_I\},t)}{\partial t} \right\rangle &= \left\langle \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \mathcal{H}(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \right\rangle, \\ i\left\langle \Phi(\{\vec{r}_i\},t) \mid \frac{\partial\Phi(\{\vec{r}_i\},t)}{\partial t} \right\rangle &= \left\langle \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \Psi(\{\vec{R}_I\},\{\vec{r}_i\},t) \right\rangle \end{aligned}$$

lead to the so-called time-dependent self-consistent-field equations [15, 16]

$$\begin{aligned} i\frac{\partial}{\partial t}\Phi(\{\vec{r}_i\},t) &= -\sum_i \frac{\nabla_i^2}{2}\Phi(\{\vec{r}_i\},t) + \left\langle \mathcal{X}(\{\vec{R}_I\},t) \mid V(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \mathcal{X}(\{\vec{R}_I\},t) \right\rangle_I \Phi(\{\vec{r}_i\},t), \\ i\frac{\partial}{\partial t}\mathcal{X}(\{\vec{R}_I\},t) &= -\sum_I \frac{\nabla_I^2}{2}\mathcal{X}(\{\vec{R}_I\},t) + \left\langle \Phi(\{\vec{r}_i\},t) \mid \mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \Phi(\{\vec{r}_i\},t) \right\rangle_i \mathcal{X}(\{\vec{R}_I\},t), \end{aligned}$$

where $\langle\rangle_I$ and $\langle\rangle_i$ are inner products with respect to $\vec{R}$ and $\vec{r}$, respectively. The next step is to approximate the nuclei as classical point particles via Wentzel-Kramers-Brillouin approximation [15, 16, 17]. The resultant Ehrenfest molecular dynamic scheme is contained in the following system

of coupled differential equations [16]:

$$M_I \frac{d^2}{dt^2}\vec{R}_I(t) = -\left\langle \Phi(\{\vec{r}_i\},t) \mid \nabla_I \mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t) \mid \Phi(\{\vec{r}_i\},t)\right\rangle_i, \tag{1.2.1}$$

$$i\frac{d}{dt}\Phi(\{\vec{r}_i\},t) = \mathcal{H}_e(\{\vec{R}_I\},\{\vec{r}_i\},t)\Phi(\{\vec{r}_i\},t). \tag{1.2.2}$$

Remark 1.2.2. Ehrenfest dynamics is a potentially powerful technique for modeling atto- to picosecond electron dynamics, but it suffers from the intrinsic problems as its wave function sibling-namely that it is implicitly based on an average potential energy surface and so does not provide state-specific information, and also suffers from problems with microscopic irreversibility.

In Ehrenfest dynamics, Newton's equation can be solved classically, whereas difficulties still arise in the simulation of Schrödinger equation due to the curse of dimensionality. The Schrödinger equation under the applied field can be written as

$$i\frac{\partial}{\partial t}\Phi(\{\vec{r}_i\},t) = \left[-\frac{1}{2}\sum_i \nabla_i^2 + \frac{1}{2}\sum_{i\neq j}\frac{1}{|\vec{r}_i - \vec{r}_j|} + \sum_i V_{ext}(\{\vec{r}_i\},t)\right]\Phi(\{\vec{r}_i\},t). \tag{1.2.3}$$

Here, one direct external potential from the electronic structure theory is the potential of electron and nuclei

$$V_{Ne}(\{\vec{R}_I\},\{\vec{r}_i\}) = -\sum_{i,I}\frac{Z_I}{|\vec{r}_i - \vec{R}_I|}.$$

For other examples, in many experiments, the laser field is applied

$$V_{laser}(\{\vec{r}_i\},t) = Ef(t)\sin(\omega t)\sum_{i=1}^{N}\vec{r}_i\cdot\vec{\alpha},$$

where α, ω, and E are respectively the polarization, the frequency, and the amplitude of the laser.

The time-dependent density functional theory (TDDFT) is commonly used to treat the Schrödinger equation in a computationally tracktable manner [18, 19, 20, 21, 22, 23, 24]. Among the different formalisms of the electronic structure theory, TDDFT achieves the excellent compromise between accuracy and efficiency. In contrast to the Schrödinger equation which relies on the "$3N+1$"-dimensional wave function, TDDFT works with the "$3+1$"-dimensional density function $\rho(\vec{r},t)$ and

therefore to achieve the dimensional reduction. The density function is obtained with the help of the fictitious system of non-interacting electrons, the Kohn-Sham system. The electrons feel an effective potential, the TDKS potential. The exact form of this potential is unknown and is extremely difficult to derive, and the approximation has to be made. As a consequence, the TDKS equation is of the form

$$\begin{aligned} i\frac{\partial}{\partial t}\psi_j(\vec{r},t) &= \left[-\frac{1}{2}\nabla^2 + V_{KS}(\vec{r},\rho,t)\right]\psi_j(\vec{r},t) \\ &= \left[-\frac{1}{2}\nabla^2 + V_{Hartree}(\rho) + V_{xc}(\rho) + V_{ext}(\vec{r},t)\right]\psi_j(\vec{r},t) \end{aligned} \tag{1.2.4}$$

where $\{\psi_j\}$ are the Kohn-Sham orbitals, V_{KS} is the Kohn-Sham potentials including the Hartree term

$$V_{Hartree}(\rho) = \iiint_V \frac{\rho(\vec{r},t)}{|\vec{r}-\vec{r}'|}\, d\vec{r}'$$

and V_{xc} is the exchange-correlation potential, which must be approximated. One simple choice is adiabatic local density approximation (LDA), which can be written as

$$V_{ext}(\rho) = C\rho^{4/3}(\vec{r},t).$$

The density function is calculated by

$$\rho = \sum_{j=1}^{N} |\psi_j|^2 .$$

Remark 1.2.3. We want to make some remarks about the model.

- The choice of exchange-correlation potential significantly affects the performance of the TDDFT. However, in the perspective of mathematics, analysis of LDA already give us much information.
- In the simulation, the pseudopotential [25, 26] is usually employed to replace the complicated effects of the motion of the core electrons of an atom and its nucleus by Coulomb interaction between electrons and nucleus. The main consideration in our perspective is that the pseudopotential reduces the number of bases set by removing the singularity. As a consequence,

this term also becomes non-stiff in the language of numerical analysis.

In TDKS equation, challenges arise for the long-time calculation. For example, in the calculations of 5TBA monolayer thin film [27], the final time is around 1 picosecond, whereas the time-step is around attosecond. In such calculations, it is difficult to obtain an accurate solution, and in addition, with well-preserved conservation laws.

The TDKS equation can be solved in many ways, perhaps the simplest of which is via explicit Runge-Kutta methods, which judiciously uses the information on the "slope" at more than one point to extrapolate the solution to the future time step. A mature package with explicit Runge-Kutta methods is well developed in [28].

Another approach is to use low-order implicit methods, such as Crank-Nicolson and Magnus expansions. One particular advantage of them over explicit methods is their stability, which allows for larger time-steps in stiff systems. In numerical analysis language, the Courant-Friedrichs-Lewy (CFL) condition is much relaxed. Progress on low-order implicit methods for the TDKS equation can be found in Octopus (http://octopus-code.org/wiki/Manual), NWCHEM (http://www.nwchem-sw.org/index.php/Main_Page), GPAW (https://wiki.fysik.dtu.dk/gpaw/) as well as other TDDFT software packages.

Splitting type algorithm is also commonly used in the community. The main reason for splitting methods over others is its support for preservation of conservation laws. Discussions of this type can be found in [29, 30].

Despite their prevalence, all those methods have the major drawback that the generalization to a stable higher-order formulation is not straightforward. As a consequence, step-size of the time is restricted by stability requirements, that the Laplacian is very stiff, and accuracy requirements, that the data can be highly oscillatory. Motivated by the challenges in those complicated simulations, we study a new class of time stepping method by spectral methods, which have been very successfully applied in the spatial direction [31, 32, 33]. The spectral method has a rich mathematical theory and is routinely used to solve the partial differential equation (PDE). One particular advantage of spectral methods over others is its high-resolution for smooth data. Cited from [34] by Trefethen, "if you want to solve an ordinary differential equation (ODE) or PDE to high accuracy on a simple domain and if the data defining the problem are smooth, then spectral methods are usually the best

tool". Since the time is only one-dimensional and there is no complex geometry involved, a natural question is to ask to what extend spectral (pseudo-spectral) methods can be likewise applied.

We consider the pseudo-spectral collocation formulation based on the Picard integral equation. There has been extensively study of this representation for ODEs [35, 36, 37]. The new representation brings with it several favorable qualities, which can be roughly summarized as follows:

- The system is well-conditioned.
- The error of the solution decays rapidly due to the spectral discretization;
- The stability of the formulation is usually not an issue;
- The global errors grow slowly;
- The quadratic first integrals are preserved while applying to the conserved system.

The pseudo-spectral collocation formulation is generally considered the optimal formulation since it has a strong mathematical foundation, and is commonly used in second-order parabolic PDEs. This implementation, however, comes at a significant expense, as the discretization by spectral methods leading to a dense system. For a N-dimensional ODE system, if m nodes are used in time, the overall size of the discretization is mN, and the computational complexity is proportional to $(mN)^3$ if Gaussian elimination is applied, which is prohibited in the large-scale calculation.

An alternative approach is to solve the discretized system iteratively. The predominant method is the deferred correction method, which is essentially applying the fixed-point iteration to a preconditioned error equation. With a simple adaptive implementation and low-order integration preconditioner, the resulting SDC [38] compares favorably with the state-of-the-art extrapolation code in moderate to high precisions. Unfortunately, when applying to the very stiff system, the performance of the deferred correction methods deteriorate and order reduction is observed. Several approaches are proposed to accelerate the convergence of deferred correction. For a discussion of the various acceleration schemes, including parareal algorithm, multigrid preconditioning, and operator splitting preconditioning, see [39, 40, 41, 42]. Among others, the Krylov subspace methods can serve as a general framework to further accelerate the convergence, by paying prices of more memory storage and overhead work. The resulting Krylov deferred correction (KDC) [43, 44], which is

recognized as applying the Newton-Krylov method to the preconditioned error equation, effectively eliminate the order reduction.

In this dissertation, we describe several new preconditioning techniques and integral equation formulations in the temporal direction that is quite general. For stiff PDEs, e.g., the TDKS equation, our algorithm is capable of optimal time step-size.

1.3 Layered media problem

In many experiments, the materials are designed by incorporating large numbers of identical inclusions (particles) in a layered material. Given an incoming source electromagnetic field, what is the scattered electromagnetic field?

Since charge and current are absent in the space and therefore, the Maxwell equation reads

$$\nabla \cdot \vec{E} = 0, \tag{1.3.1}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}, \tag{1.3.2}$$

$$\nabla \cdot \vec{B} = 0, \tag{1.3.3}$$

$$\nabla \times \vec{B} = \frac{1}{c^2}\frac{\partial \vec{E}}{\partial t}. \tag{1.3.4}$$

They constitute a set of coupled, first-order, PDEs for $\vec{E}$ and $\vec{B}$. Applying the curl to the Eq. (1.3.2),

$$(\nabla \cdot \vec{E}) - \nabla^2 \vec{E} = -\frac{\partial}{\partial t}(\nabla \times \vec{B}).$$

With the help of Eqs. (1.3.1) and (1.3.4), it can be derived that

$$\nabla^2 \vec{E} = \frac{1}{c^2}\frac{\partial^2 \vec{E}}{\partial t^2}.$$

In much of same way, applying the curl to Eq. (1.3.4),

$$\nabla(\nabla \cdot \vec{B}) - \nabla^2 \vec{B} = \frac{1}{c^2}\frac{\partial}{\partial t}(\nabla \times \vec{E}).$$

With the help of Eqs. (1.3.3) and (1.3.2), it can be derived that

$$\nabla^2 \vec{B} = \frac{1}{c^2} \frac{\partial^2 \vec{B}}{\partial t^2}.$$

Now separate equations for $\vec{E}$ and $\vec{B}$ are obtained. In vacuum, then, each Cartesian component of $\vec{E}$ and $\vec{B}$ satisfies the three-dimensional wave equation,

$$\nabla^2 f = \frac{1}{v^2} \frac{\partial^2 f}{\partial t^2}.$$

As discussed in [45], a dielectric medium that is linear, isotropic, homogeneous, and nondispersive, all components of the electric and magnetic field behave identically and their behaviors are fully described by a single scalar wave equation. We then turn away from the vector theory to the simple scalar theory. By separation of variables, the wave equation can be simplified. Assume that

$$f(\vec{x}, t) = A(\vec{x})T(t).$$

Substitute this into the wave equation, we obtain

$$\frac{\nabla^2 A}{A} = \frac{1}{v^2 T} \frac{d^2 T}{dt}.$$

This equation is valid only when both sides are constant. By assuming the constant be $-\omega^2$ without loss of generality, we have

$$\begin{aligned} \frac{\nabla^2 A}{A} &= -\omega^2, \\ \frac{1}{c^2 T} \frac{d^2 T}{dt^2} &= -\omega^2. \end{aligned}$$

Rearranging the first equation, the Helmholtz equation is obtained

$$(\nabla^2 + \omega^2) A = 0.$$

Likewise, let $k = \omega v$, the second equation becomes

$$\left(\frac{d^2}{dt^2} + k^2\right) T = 0.$$

The solution in time will be a linear combination of sine/cosine functions and the solution in space will depend on the boundary conditions. Hence, by appropriate assumptions, the wave equation can be simplified to the Helmholtz equation. In layered media problems, complicate boundary conditions in each layer are imposed.

Three commonly used methods to approximate the Helmholtz equation in a media are finite difference methods [46], finite element methods [47], and boundary integral methods [48]. In the finite difference method, the region of interest is discretized by a set of points at which the differential operator is approximated by Taylor expansion. For a typical finite element method, the differential equation is recast as a weak formulation, and the computational domain is partitioned into a finite number of small elements in which low-order polynomials approximations are used.

Despite their prevalence, both finite difference and finite elements have the major drawback that when applying to the infinite domain, the entire volume must be discretized with artificial truncation [49, 50]. In contrast, integral equations in this circumstance often require unknown only on the domain boundary, leading to a dimensional reduction in the linear system to be solved. In this sense, integral equation reveals the true size of a problem. This is one of the reasons that we have adopted an integral equation approach. Furthermore, the discretization of the integral equation formulation leads to a well-conditioned linear system. Specifically, the condition number of the corresponding matrix A typically is a constant, which enables more stable convergence. On the other hand, the condition number of finite difference and finite element methods grow as $\mathcal{O}(\frac{1}{h^2})$, where h is the minimum mesh width.

For a review of recent developments in numerical methods for the Helmholtz equation, including finite difference, finite element, and integral equation methods, see [51].

We now turn to a brief treatment of integral equations of Helmholtz equation, using two-layered media problem as an example. For a complete account, including potential and Fredholm theory, we refer the interested reader to [52, 53, 54].

Consider the two-dimensional problem of computing the scattered field due to a unit-strength

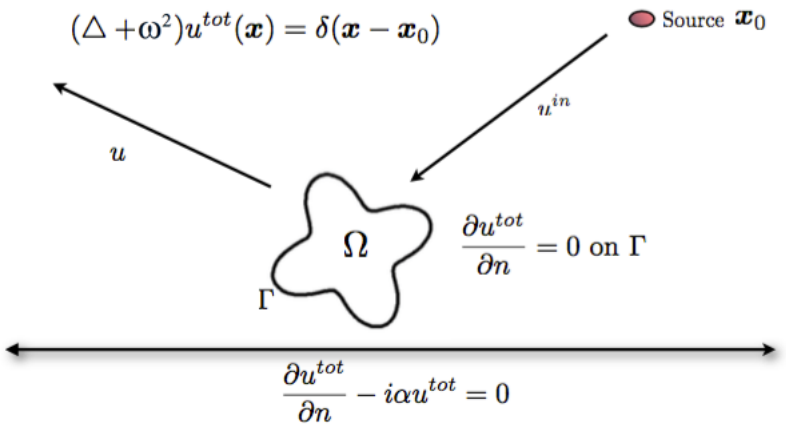


Figure 1.1: Scattering from a sound-hard obstacle above an impedance plane.

point source located at $\vec{x}_0 = (x_0, y_0)$ in the presence of a "sound-hard" obstacle over an infinite half-space subject to impedance boundary conditions in Figure 1.3. Let the total field be defined as $u^{tot} = u^{in} + u$, where u^{in} denotes the (known) incoming field due to the point source and u denotes the scattered field. Since the scattered field involves no sources outside the obstacle Ω, it must satisfy the homogeneous Helmholtz equation

$$(\Delta + \omega^2)u(\vec{x}) = 0.$$

On the obstacle boundary Γ, the total field must satisfy homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial n} = -\frac{\partial u^{in}}{\partial n},$$

where $\frac{\partial}{\partial n}$ is the outward normal derivative. Finally, on the interface, an impedance condition is imposed such that

$$\frac{\partial u^{tot}}{\partial n} - i\alpha u^{tot} = 0.$$

By integral equation methods, an ansatz is to represent the total field as

$$u^{tot}(\vec{x}) = \int_\Gamma g_{\omega,\alpha}(\vec{x}, \vec{y})\sigma(\vec{y})\ d\vec{y} + u^{in}(\vec{x}),$$

where $g_{\omega,\alpha}$ is the domain Green's function, i.e., the Green's function of half-space with homogeneous impedance boundary conditions, and $u^{in}(\vec{x}) = g_{\omega,\alpha}(\vec{x}, \vec{x}_0)$. Imposing the Neumann boundary condition on the obstacle yields that Fredholm integral equation of the second kind:

$$-\frac{1}{2}\sigma(\vec{x}) + \int_\Gamma \frac{\partial}{\partial n_x} g_{\omega,\alpha}(\vec{x}, \vec{y})\sigma(\vec{y})\ d\vec{y} = -\frac{\partial}{\partial n_x} g_{\omega,\alpha}(\vec{x}, \vec{x}_0)$$

for $\vec{x} \in \Gamma$. To solve the integral equation efficiently, a typical approach is to use the iterative solver, such as generalized minimal residual method (GMRES) [55, 56], with the acceleration of fast convolution technique, such as fast multipole method (FMM) [57]. While there exist many existing methods for the fast evaluation of the free-space Green's function, however, the question remains

open that how to evaluate the domain Green's function efficiently. The domain Green's function is so complicated that the integral form is usually used to represent it. Satisfying the Sommerfeld radiation condition

$$\lim_{r\to\infty}\sqrt{r}\left(\frac{\partial}{\partial r}g_{\omega,\alpha}-i\omega g_{\omega,\alpha}\right)=0$$

the domain Green's function can be written in the Sommerfeld representation

$$g_{\omega,\alpha}(\vec{x})=\frac{1}{4\pi}\int_{-\infty}^{\infty}\frac{e^{-\sqrt{\lambda^2-\omega^2}(y+y_0)}}{\sqrt{\lambda^2-\omega^2}}e^{i\lambda(x-x_0)}\left(\frac{\sqrt{\lambda^2-\omega^2}+i\alpha}{\sqrt{\lambda^2-\omega^2}-i\alpha}\right)\,d\lambda$$

or in the form of

$$g_{\omega,\alpha}(\vec{x})=g_k(\vec{x},\vec{x}_0-2y_0\widehat{y})+2i\alpha\int_0^{\infty}g_k(\vec{x},\vec{x}_0-(2y_0+\eta)\widehat{y})e^{i\alpha\eta}\,d\eta$$

with the help of method of images. Here, $\vec{x}_0=(x_0,y_0)$ is the location of the source, g_k is the free-space Green's function, and $\widehat{y}=(0,1)$.

The main contribution previously along this direction has been focusing on compress the discretization of the integral form of the domain Green's function [58, 59, 60, 61, 62, 63, 64, 65, 66]. Unfortunately, this approach is considered inefficient as the data is too complicated to compress. In this dissertation, the free-space Green's function is compressed, the modified translation formulas are derived, and then the information is translated in the hierarchical tree structure in the fast multipole fashion. One striking feature of our method is its efficiency: the computational complexity is similar to the evaluation of Green's function in free space and the algorithm can be orders of magnitude faster than existing schemes for simulating 2-D waves in two-layered media.

1.4 Outline of the dissertation

The remainder of this dissertation is organized as follows.

In Chapter 2, we give a complete description of SDC type algorithms, starting with the pseudo-spectral collocation formulation of ODE. We then give deferred correction algorithms, which are preconditioned iterative methods, for the fast convergence of the solution. In particular, we discuss our new collocation formulation by Green's function and several new preconditioning techniques. This is followed by the derivation and convergence analysis, with extensive numerical experiments.

In Chapter 3, we discuss applications of the hierarchical methods. We begin with the parallel version of the fast hierarchical solver for two-point boundary value problem, which has been used as a preconditioner for SDC. Then we introduce a new hierarchical method for layered media problem. This is followed by complexity and error analysis, which we verify through extensive numerical experiments.

Finally, in Chapter 4, we end with some generalizations and concluding remarks, including some new directions of “optimal" preconditioning technique, a generalization of the integral equation method (IEM), and prescriptions on adapting our hierarchical methods to multi-layered media problems. These are all characterized by immense search spaces. The dissertation concludes with a summary of our main results and contributions.

CHAPTER 2

Advanced Iterative Methods in Time [1]

2.1 Introduction

The construction of efficient, stable, and high-order methods for solving the IVP is, in many respects, a mature subject. For ODE IVPs, the linear multistep methods and Runge-Kutta methods have been extensively studied in both theory and implementation [67, 68, 69, 70]. Widely used ODE IVP solvers include the backward differentiation formula based DASPK [71, 72] and Runge-Kutta method based Radau5 [73]. We refer interested readers to [74] for existing theoretical results, different algorithms, and software packages. Many of these numerical simulation tools have been successfully applied in research studies and have significantly advanced our knowledge in science and engineering. However, these advances in turn also revealed the limitations of existing numerical algorithms, which can be roughly divided into two classes of systems:

- chaotic system: one example is the simulation of Kuramoto-Sivashinsky equation [75, 76], which models thermal diffusive instabilities in laminar flame fronts. The solution of PDE can exhibit chaotic behaviors. For instance, the perturbation of initial data can be amplified by as much as 10^8 up to time $t = 150$. Given the lack of confidence of existing methods, it is important to obtain a very accurate result to study and explore the chaotic solutions in the systems. However, the generalization of the existing methods from low-order to higher-order, especially in stiff systems, is not straightforward.

- long-time dynamic: for instance, difficulties arise in the simulations of electron dynamics with applications in the electronic structure theory by real-time time-dependent density functional

[1] Some materials in this chapter previously appeared as an article in the Journal of Scientific Computing. The origin al citation is as follows: W. Qu, N. Brandon, D. Chen, J. Huang, and T. Kress. "A numerical framework for integrating deferred correction methods to solve high order collocation formulations of ODEs", 68 (2), 484-520, 2016. Some materials in this chapter will appear in the future and are currently in preparation.

theory [18, 77, 1, 2]. In this simulation, scientists are interested in the phenomena occurred in the scale of picosecond (10^{-12} second) or even longer, whereas the time scale of the system is attosecond (10^{-18} second). There are several difficulties in simulating such long-time dynamics using the existing algorithm, including the stability restriction (CFL condition) and accuracy restriction (due to low-order) of the step-size, as well as the fast growth rate of the global error.

Our primary goal is to improve the performance of the algorithm in these regards.

In this chapter, we study a new class of time-stepping method by (pseudo-) spectral methods. We begin by converting the original ODE into the corresponding Picard equation, which has several analytical advantages. However, the discretization leads to a large dense matrix. The standard direct method, such as Gaussian elimination will be computationally intractable in the large-scale calculation. To efficiently solve the system, iterative methods such as deferred correction are proposed. The resulting SDC has been successfully applied to many problems and have shown advantages in the high-accuracy regime. Unfortunately, when applying to the stiff system, the performance of the deferred correction methods deteriorate and order reduction is observed. Several approaches are proposed to improve the performance of the deferred correction scheme, which can be classified, loosely speaking, into three groups.

- Low-order methods: the first group seeks different low-order preconditioners to have a better-conditioned system or to reduce the computational complexity. In [40], higher-order preconditioners based on Runge-Kutta with uniform grids are proposed and analysis indicates that preconditioners with uniform sampling have better performance; in [78], operator splitting methods are served as the preconditioner and show some advantages; in [42], the alternative direction implicit (ADI) approach is used as a preconditioner in parabolic PDEs with linear complexity; in [79], multigrid methods are used in the temporal direction to reduce the computational cost.

- Parallelization: in the past decades, we have seen an increase in research into the parallelization of numerical methods for time-dependent differential equations beginning with the introduction of the parareal algorithm [80, 81] and the related parallel implicit time-integrator scheme in 2003 [82]. Then in 2008, Michael Minion and his collaborators bring the parareal idea into the SDC and a significant decrease in the overall computational cost is obtained in the preliminary

examples [83].

- Krylov subspace methods: by re-formulating the SDC in the framework of linear algebra, the deferred correction approach can be recognized as applying the fixed-point iteration to a preconditioned error equation. In particular, for linear problems, the deferred correction is equivalent to a Neumann series, which motivates researchers to use the Krylov subspace method to accelerate the convergence. The resulting KDC [43, 44] effectively eliminate the order reduction with more memory usage and some overhead calculations.

In this dissertation, we develop new mathematical and computational techniques for the accurate and efficient solutions by collocation schemes. First, motivated by the success of parareal idea in the temporal direction, we introduce a new class of diagonal preconditioner. Different from the parareal SDC, which performed over multiple time-steps in parallel, the diagonal preconditioner aim to parallelize the function evaluation in every single local interval. Furthermore, our analysis indicates that the diagonal preconditioner is more stable than the traditional triangular preconditioner. Preliminary results suggest that the diagonal preconditioner is capable of achieving high parallel efficiency and show some advantages in the modern computer architecture.

Secondly, we have noticed there exist many powerful low-order preconditioners and each of them has certain advantages and disadvantages. A natural question is to ask how to choose a preconditioner with optimal performance under certain conditions. To improve the performance of the deferred correction, we propose a preconditioning technique, which combines different preconditioners, to train an “optimal" preconditioner to optimize the conditioning of the preconditioned system. The optimization procedure can be calculated in the precomputation stage. As a result, the iteration number by deferred correction with “optimal" preconditioner is reduced significantly.

Thirdly, we propose a new formulation to “remove" the stiffness in the stiff systems. In many applications, the equation consists stiff linear operators and non-stiff nonlinear operators. Examples include Navier-Stokes equation, nonlinear Schrödinger equation, diffusion equation and so on. For this class of models, we use the Green’s function of the linear operator to have a well-conditioned system. When the stiffness is “removed", the equation becomes non-stiff and we can apply explicit methods to efficiently solve the system. Comparing to the SDC, the performance by the new approach, IEM, is much improved.

The chapter is organized as follows. Section 2.2 briefly reviews the spectral and Krylov deferred correction. The performance of the deferred correction is analyzed in Section 2.3. Our new preconditioning techniques diagonal preconditioners, “optimal" preconditioners, and IEM are introduced in Section 2.4, 2.5, and 2.6, respectively. Finally, in Section 2.7, the application to the TDDFT is discussed. Materials in the Section 2.3 has been presented in [84] and papers reporting on Section 2.4-2.6 is in preparation [85, 86].

2.2 Spectral/Krylov deferred correction

In this section, we briefly review the SDC [38] and KDC [43]. We begin by discussing the collocation formulation of the differential equation with rigorous analysis. Then the deferred correction scheme is described to solve the system. The perspective from linear algebra is provided for better understanding of the algorithm. Finally, we discuss how the Krylov subspace methods can be used to accelerate the convergence of the deferred correction.

2.2.1 Collocation formulation

For simplicty, consider the 1-dimensional ODE of the form

$$\begin{cases} y'(t) = f(t,y), \quad t \in [0,T], \\ y(0) = y_0 \end{cases} \tag{2.2.1}$$

where y and f are assumed to be smooth, which is the requirement for high-order construction. In addition, we focus on the single interval with refinement $t_0, t_1, \ldots, t_m, t_{m+1}$ such that

$$\begin{aligned} t_0 &= 0, & t_{m+1} &= T, \\ t_0 &\le t_1 < t_2 < \cdots < t_m \le t_{m+1}. \end{aligned}$$

We use Gauss-Legendre points $t_1, \ldots, t_m$ in time, unless otherwise stated. We also denote Δt as the length of the interval $[0,T]$ and Δt_n as $t_{n+1} - t_n$.

The idea of pseudo-spectral collocation formulation can be chased back in the 1960s [87, 88, 89]. However, the original algorithm was not be very popular since it directly discretized the differential operator, which is ill-conditioned numerically. To avoid the problem, in [38], the differential equation

is recast as a Picard integral equation,

$$y(t) = y_0 + \int_0^t f(\tau, y)\ d\tau.$$

The integral equation formulation gives us a well-conditioned system as the numerical integration is more stable than numerical differentiation.

Remark 2.2.1. The maximum of spectral differentiation grows like $\mathcal{O}(m^2)$, whereas the maximum eigenvalue of spectral integration matrix is bounded, and the minimum eigenvalue decays like $\mathcal{O}(\frac{1}{m^2})$.

For discretization, first we consider the **y-formulation**. Since f smooth, given the values of $\widetilde{y}$, we interpolate it with a high-order orthonormal polynomial. In Lagrangian polynomial form, we write

$$\widetilde{f}(t, \widetilde{y}) = \sum_{j=1}^{m} \ell_j(t) \widetilde{f}(t_j, \widetilde{y}),$$

where

$$\ell_j(t) = \prod_{i \neq j} \frac{t - t_i}{t_j - t_i}.$$

Then the solution of $\widetilde{y}$ is recovered by the integral

$$\widetilde{y}(t) = y_0 + \int_0^t \widetilde{f}(\tau, \widetilde{y})\ d\tau.$$

In order to uniquely determine the polynomials, the equation is required to be satisfied at each collocation points and we refer it as collocation formulation.

In some differential algebraic equation (DAE) and PDE applications, sometimes it is more convenient to directly work with the unknown variables y_t [90]. To do so, y_t is interpolated as

$$\widetilde{y}_t(t) = \sum_{j=1}^{m} \ell_j(t) \widetilde{y}_t(t_j)$$

and therefore the solution is obtained by integral

$$\widetilde{y}(t) = y_0 + \int_0^t \widetilde{y}_\tau(\tau)\ d\tau.$$

To determine the polynomial, the equation

$$\widetilde{y}_t(t) = f\left(t, y_0 + \int_0^t \widetilde{y}_\tau(\tau)\ d\tau\right) \tag{2.2.2}$$

is to be satisfied at collocation points. We refer this as **yp-formulation**. The collocation formulation has several advantages, which are summarized in the following theorem provided by Hairer:

Theorem 2.2.1 (Hairer [35])**.** *For ODE IVPs, the Gauss collocation formulation in Eq.* (2.2.2) *with* m *nodes is of order* $2m$ *(super convergence), A-stable, B-stable, symplectic (structure preserving), and symmetric (time-reversible). In addition, the error decays exponentially with* m *increases.*

With a little effort, we are able to show that the same results should hold for y-formulation.

Theorem 2.2.2. *The same result holds if we replace the yp-formulation with the y-formulation in Theorem 2.2.1.*

Proof. It is sufficient to show that by y-formulation, the collocation formulation also holds for the yp-formulation. □

For the rest of materials, we will focus on the y-formulation, unless otherwise stated. In the theorem, the superconvergence and exponentially decay properties ensure the accurate solution; A-stable, B-stable imply that the stability is usually not the issue of the collocation formulation; symplecticness ensures that the conservation laws are well preserved in the conserved system. The next theorem provided by Hairer can help us to gain more insight into the long-time simulation.

Theorem 2.2.3 (Hairer [35])**.** *Consider a completely integrable Hamiltonian system*

$$\begin{cases} \dot{p} &= -\nabla_q H(p,q), \\ \dot{q} &= \nabla_p H(p,q) \end{cases}$$

with real analytic Hamiltonian. We let $(p, q) = \psi(a, \theta)$ *be the symplectic diffeomorphism that transform the Hamiltonian equation to action-angle variables, and we denote the inverse transformation by* $(a, \theta) = (I(p, q), \Theta(p, q))$. *Consequently, the components* $I_1, \ldots, I_d$ *of* I *are first integrals of the system. In the action-angle variables, the Hamiltonian is* $K(a) = H(p, q)$, *and we denote the vector of frequencies by* $\omega(a) = \nabla K(a)$. *We consider this in a neighbourhood of some* $a^* \in \mathbb{R}^d$. *If we apply the symplectic integrator of order* p *with globally defined modified Hamiltonian with strong non-resonance condition for* $\omega(a^*)$ *that*

$$|k \cdot \omega(a^*)| \geq \gamma |k|^{-v}, \ k \in \mathbb{Z}^d, \ k \neq 0$$

and the condition

$$\|I(p_0, q_0) - a^*\| \leq Const |\log(h)|^{-v-1}.$$

Then, there exist constants C, h_0 *such that for* $h \leq h_0$ *and for* $t = nh \leq h^{-p}$ *the numerical solution satisfies*

$$\|(p_n, q_n) - (p(t), q(t))\| \leq Cth^p,$$
$$\|I(p_n, q_n) - I(p_0, q_0)\| \leq Ch^p.$$

The theorem shows that the collocation formulation has the particular advantage in the long-time simulation for a completely integrable Hamiltonian system. The example adopted from [35] is provided to compare the collocation formulation with widely used Runge-Kutta.

Example 2.2.1 (Toda Lattice)**.** This is a system of particles on a line interacting pairwise with exponential forces. The motion is determined by the Hamiltonian

$$H(p, q) = \sum_{k=1}^{n} \left(\frac{1}{2} p_k^2 + e^{q_k - q_{k+1}} \right)$$

with period boundary conditions $q_{n+1} = q_1$. With the notation $a_k = -\frac{1}{2} p_k$, $b_k = \frac{1}{2} e^{\frac{1}{2}(q_k - q_{k+1})}$, all n

eigenvalues of the matrix

$$L = \begin{bmatrix} a_1 & b_1 & & & b_n \\ b_1 & a_2 & b_2 & 0 & \\ & b_2 & \ddots & \ddots & \\ & 0 & \ddots & a_{n-1} & b_{n-1} \\ b_n & & & b_{n-1} & a_n \end{bmatrix}$$

are first integrals of the system. It can be shown that the system is completely integrable. We consider the case when $n = 3$ with initial conditions $p_1 = -1.5, p_2 = 1, p_3 = 0.5$, and $q_1 = 1, q_2 = 2, q_3 = -1$ marching to time $T = 100$. We compare the explicit fourth-order Runge-Kutta method (RK4) with symplectic integrator collocation formulation with 5 Gauss nodes.

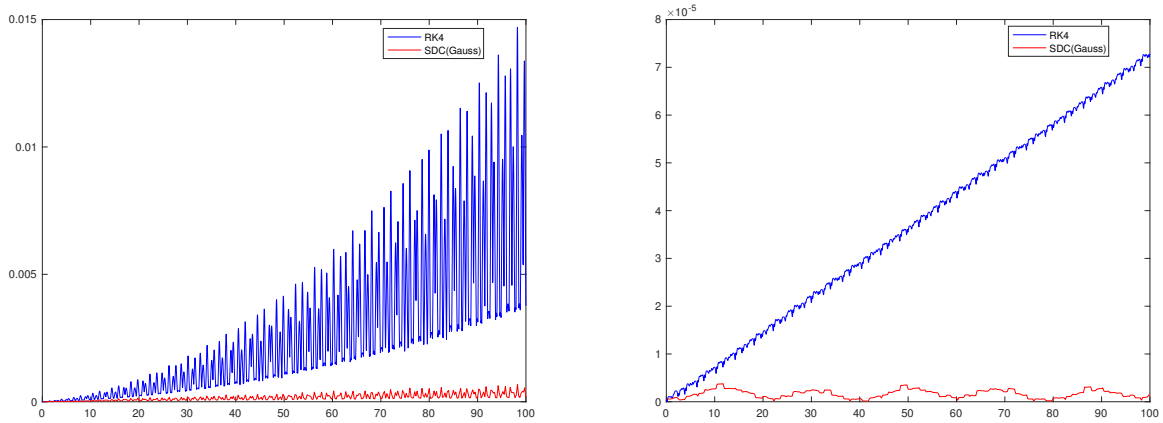


Figure 2.1: Errors of solutions and first integrals for RK4 and Gauss collocation formulation method

From Figure 2.1, we see that the error of the solution by RK4 grows quadratically, where the error of the collocation formulation only grows linearly. For the conservation law, the error by RK4 grows linearly, whereas the error by collocation formulation doesn't seem to grow.

2.2.2 Deferred correction

Although mathematical properties of the collocation formulation are very attractive, the direct method is too expensive to apply in large systems. As a result, deferred correction method is commonly used.

Assume we have a provisional solution $\widetilde{y}$, the error is measured by the residual function

$$\epsilon(t) := y_0 + \int_0^t \widetilde{f}(\tau, \widetilde{y})\ d\tau - \widetilde{y}(t). \tag{2.2.3}$$

Define the error function as

$$\delta(t) := y(t) - \widetilde{y}(t).$$

Restricting the error function in finite dimensional polynomial space, then we have the collocation formulation of the error equation

$$\widetilde{\delta}(t) = \int_0^t \widetilde{f}(\tau, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(\tau, \widetilde{y})\ d\tau + \epsilon(t). \tag{2.2.4}$$

By some algebra, locally we have

$$\widetilde{\delta}(t_{n+1}) = \widetilde{\delta}(t_n) + \int_{t_n}^{t_{n+1}} \widetilde{f}(\tau, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(\tau, \widetilde{y})\ d\tau + \epsilon(t_{n+1}) - \epsilon(t_n).$$

To solve the error equation efficiently, low-order methods are used for approximation. For stiff system, implicit methods, such as backward Euler's method is used so that

$$\widetilde{\delta}(t_{n+1}) \approx \widetilde{\delta}(t_n) + \Delta t_n[\widetilde{f}(t_{n+1}, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_{n+1}, \widetilde{y})] + \epsilon(t_{n+1}) - \epsilon(t_n). \tag{2.2.5}$$

By linear implicit approximation, Eq. (2.2.5) can be solved by

$$\widetilde{\delta}(t_{n+1}) \approx \widetilde{\delta}(t_n) + \Delta t_n \frac{\partial \widetilde{f}(t_{n+1}, \widetilde{y})}{\partial y} \widetilde{\delta}(t_{n+1}) + \epsilon(t_{n+1}) - \epsilon(t_n). \tag{2.2.6}$$

For non-stiff system, we use explicit methods, such as forward Euler's method,

$$\widetilde{\delta}(t_{n+1}) \approx \widetilde{\delta}(t_n) + \Delta t_n[\widetilde{f}(t_n, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_n, \widetilde{y})] + \epsilon(t_{n+1}) - \epsilon(t_n). \tag{2.2.7}$$

The error is added to the provisional solution until the certain threshold is reached.

Remark 2.2.2. It is possible to update the error with higher-order methods. For example, with

Trapezoidal's rule, we have

$$\widetilde{\delta}(t_{n+1}) \approx \widetilde{\delta}(t_n) + \frac{\Delta t_n}{2}[(\widetilde{f}(t_n, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_n, \widetilde{y})) + (\widetilde{f}(t_{n+1}, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_{n+1}, \widetilde{y})] + \epsilon(t_{n+1}) - \epsilon(t_n).$$

However, the performance of deferred correction by higher-order preconditioners is not necessarily better. We will discuss it later.

2.2.3 Perspective of linear algebra

To explore the properties of deferred correction and improve its performance, it is helpful to reformulate the problem in terms of linear algebra language. First, we give some notations, denote

$$\begin{aligned}
\mathbf{Y}_0 &= \begin{bmatrix} y(0) & \cdots & y(0) \end{bmatrix}^T, \\
\mathbf{Y} &= \begin{bmatrix} \widetilde{y}(t_1) & \cdots & \widetilde{y}(t_m) \end{bmatrix}^T, \\
\mathbf{F}(\mathbf{Y}) &= \begin{bmatrix} \widetilde{f}(t_1, \widetilde{y}) & \cdots & \widetilde{f}(t_m, \widetilde{y}) \end{bmatrix}^T.
\end{aligned}$$

In addition, define the normalized spectral integral matrix S such that

$$\Delta t[S]_{i,j} = \int_0^{t_i} \ell_j(s)\ ds,$$

where the integral can be analytically precomputed. Then we have

$$\Delta t S(i,:) \cdot \mathbf{F}(\mathbf{Y}) = \int_0^{t_i} \widetilde{f}(\tau, \widetilde{y})\ d\tau.$$

Here, we use the MATLAB notation such that $S(i,:)$ indicates the ith row of the matrix S.

Similarly, we define the lower-triangular integration matrix for Euler's method. For backward

Euler's method, we have

$$\Delta t\widetilde{S} = \begin{bmatrix} \Delta t_0 & 0 & 0 & \dots & 0 \\ \Delta t_0 & \Delta t_1 & 0 & \dots & 0 \\ \Delta t_0 & \Delta t_1 & \Delta t_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta t_0 & \Delta t_1 & \Delta t_2 & \dots & \Delta t_{m-1} \end{bmatrix}.$$

Acting the integration matrix on the function gives us the right-hand rule of the Riemann sum

$$\Delta t\widetilde{S}(i,:) \cdot \mathbf{F}(\mathbf{Y}) = \sum_{j=0}^{i-1} \Delta t_j \widetilde{f}(t_{j+1}, \widetilde{y}).$$

For forward Euler's method, we have

$$\widetilde{S} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \Delta t_1 & 0 & 0 & \dots & 0 \\ \Delta t_1 & \Delta t_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta t_1 & \Delta t_2 & \dots & \Delta t_{m-1} & 0 \end{bmatrix}.$$

Acting the integration matrix on the function gives us the left-hand rule of the Riemann sum

$$\Delta t\widetilde{S}(i,:) \cdot \mathbf{F}(\mathbf{Y}) = \sum_{j=1}^{i-1} \Delta t_j \widetilde{f}(t_j, \widetilde{y}).$$

With the setup, the collocation formulation can be written in the matrix form

$$\mathbf{Y} = \mathbf{Y}_0 + \Delta t S \mathbf{F}(\mathbf{Y}). \tag{2.2.8}$$

To solve the collocation equation is equivalent to the following root-finding problem:

$$\boldsymbol{\epsilon} = \mathbf{Y}_0 + \Delta t S \mathbf{F}(\mathbf{Y}) - \mathbf{Y}.$$

Denote the provisional solution as $\widetilde{\mathbf{Y}}$ and define the error function as

$$\boldsymbol{\delta} = \mathbf{Y} - \widetilde{\mathbf{Y}}.$$

Substitute the relationship into the equation, we have

$$\boldsymbol{\delta} = \Delta t S[\mathbf{F}(\widetilde{\mathbf{Y}} + \boldsymbol{\delta}) - \mathbf{F}(\widetilde{\mathbf{Y}})] + \mathbf{Y}_0 + \Delta t S\mathbf{F}(\widetilde{\mathbf{Y}}) - \widetilde{\mathbf{Y}}.$$

Directly solving the equation is very expensive, therefore the low-order approximation is used

$$\widetilde{\boldsymbol{\delta}} = \Delta t \widetilde{S}[\mathbf{F}(\widetilde{\mathbf{Y}} + \widetilde{\boldsymbol{\delta}}) - \mathbf{F}(\widetilde{\mathbf{Y}})] + \mathbf{Y}_0 + \Delta t S\mathbf{F}(\widetilde{\mathbf{Y}}) - \widetilde{\mathbf{Y}}. \tag{2.2.9}$$

Define the function $\widetilde{\mathbf{H}}(\widetilde{\mathbf{Y}})$ as by solving the approximated error function with provisional solution $\widetilde{\mathbf{Y}}$, the deferred correction approach can be viewed as finding the roots of the nonlinear system

$$\widetilde{\mathbf{H}}(\widetilde{\mathbf{Y}}) = \mathbf{0}$$

by a fixed point iteration

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + \widetilde{\mathbf{H}}(\mathbf{Y}^{[n]})$$

. Applying the implicit function theorem, it can be verified that the system is preconditioned

$$\mathbf{J}_{\widetilde{\mathbf{H}}}(\widetilde{\mathbf{Y}}) = -(I - \Delta t \widetilde{S}\mathbf{J}_{\mathbf{F}}(\widetilde{\mathbf{Y}}))^{-1}(I - \Delta t S\mathbf{J}_{\mathbf{F}}(\widetilde{\mathbf{Y}})).$$

Since the preconditioner $I - \Delta t \widetilde{S}\mathbf{J}_{\mathbf{F}}(\widetilde{\mathbf{Y}})$ has the low-triangular shape, we refer this as the **triangular preconditioner**.

Remark 2.2.3. Solving the linear system (2.2.9) is identical to the correction procedure we discussed

in Eqs. (2.2.5) and (2.2.7). By using the forward Euler's method, we have the linear system

$$\begin{bmatrix} \widetilde{\delta}(t_1) \\ \widetilde{\delta}(t_2) \\ \widetilde{\delta}(t_3) \\ \vdots \\ \widetilde{\delta}(t_m) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \Delta t_1 & 0 & 0 & \dots & 0 \\ \Delta t_1 & \Delta t_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta t_1 & \Delta t_2 & \dots & \Delta t_{m-1} & 0 \end{bmatrix} \begin{bmatrix} \widetilde{f}(t_1, \widetilde{y}(t_1) + \widetilde{\delta}(t_1)) - \widetilde{f}(t_1, \widetilde{y}(t_1)) \\ \widetilde{f}(t_2, \widetilde{y}(t_2) + \widetilde{\delta}(t_2)) - \widetilde{f}(t_2, \widetilde{y}(t_2)) \\ \widetilde{f}(t_3, \widetilde{y}(t_3) + \widetilde{\delta}(t_3)) - \widetilde{f}(t_3, \widetilde{y}(t_3)) \\ \vdots \\ \widetilde{f}(t_m, \widetilde{y}(t_m) + \widetilde{\delta}(t_m)) - \widetilde{f}(t_m, \widetilde{y}(t_m)) \end{bmatrix} + \begin{bmatrix} \epsilon(t_1) \\ \epsilon(t_2) \\ \epsilon(t_3) \\ \vdots \\ \epsilon(t_m) \end{bmatrix}.$$

From here, we can easily recognize the error function can be solved by Eq. (2.2.7) from t_1 to t_m. Similarly, using the backward Euler's method gives us the linear system

$$\begin{bmatrix} \widetilde{\delta}(t_1) \\ \widetilde{\delta}(t_2) \\ \widetilde{\delta}(t_3) \\ \vdots \\ \widetilde{\delta}(t_m) \end{bmatrix} = \begin{bmatrix} \Delta t_1 & 0 & 0 & \dots & 0 \\ \Delta t_1 & \Delta t_2 & 0 & \dots & 0 \\ \Delta t_1 & \Delta t_2 & \Delta t_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta t_1 & \Delta t_2 & \dots & \Delta t_{m-1} & \Delta t_m \end{bmatrix} \begin{bmatrix} \widetilde{f}(t_1, \widetilde{y}(t_1) + \widetilde{\delta}(t_1)) - \widetilde{f}(t_1, \widetilde{y}(t_1)) \\ \widetilde{f}(t_2, \widetilde{y}(t_2) + \widetilde{\delta}(t_2)) - \widetilde{f}(t_2, \widetilde{y}(t_2)) \\ \widetilde{f}(t_3, \widetilde{y}(t_3) + \widetilde{\delta}(t_3)) - \widetilde{f}(t_3, \widetilde{y}(t_3)) \\ \vdots \\ \widetilde{f}(t_m, \widetilde{y}(t_m) + \widetilde{\delta}(t_m)) - \widetilde{f}(t_m, \widetilde{y}(t_m)) \end{bmatrix} + \begin{bmatrix} \epsilon(t_1) \\ \epsilon(t_2) \\ \epsilon(t_3) \\ \vdots \\ \epsilon(t_m) \end{bmatrix}.$$

It is also recognized that the error equation can be solved by Eq. (2.2.5).

Remark 2.2.4. We consider a general case how Eq. (2.2.9) can be solved for PDEs. Consider the 1-dimensional parabolic PDE of the form

$$u_t(x,t) = a(x)u_{xx}(x,t) + b(x)u_x(x,t) + c(x)u(x,t) + f(x,t)$$

with appropriate initial and boundary conditions. Integrate it with time, we have

$$u(x,t) = u_0(x) + \int_0^t a(x)u_{xx}(x,s) + b(x)u_x(x,s) + c(x)u(x,s) + f(x,s)\ ds$$

With approximated solution $\widetilde{u}(x,t)$, we define the residual function as

$$\epsilon(x,t) = u_0(x) + \int_0^t a(x)\widetilde{u}_{xx}(x,s) + b(x)\widetilde{u}_x(x,s) + c(x)\widetilde{u}(x,s) + f(x,s)\ ds - \widetilde{u}(x,t).$$

Then the error equation can be derived to satisfy

$$\delta(x,t) = \int_0^t a(x)\delta_{xx}(x,s) + b(x)\delta_x(x,s) + c(x)\delta(x,s)\ ds$$

with zero initial condition and also some kind of zero boundary conditions if exact boundary conditions are used. Restricting the solution in the finite-dimensional polynomial space, the collocation of error equation can be written as

$$\widetilde{\delta}(x,t) = \int_0^t a(x)\widetilde{\delta}_{xx}(x,s) + b(x)\widetilde{\delta}_x(x,s) + c(x)\widetilde{\delta}(x,s)\ ds$$

By backward Euler's method, in each iteration, we have to solve

$$\widetilde{\delta}(x,t_{n+1}) = \widetilde{\delta}(x,t_n) + a(x)\widetilde{\delta}_{xx}(x,t_{n+1}) + b(x)\widetilde{\delta}_x(x,t_{n+1}) + c(x)\widetilde{\delta}(x,t_{n+1}).$$

A fast recursive solver can be employed with a capability of general boundary conditions to efficiently solve the equation adaptively. We leave the details of the algorithm in Chapter 3.1.

2.2.4 Krylov deferred correction

In the stiff problems, the order reduction becomes very severe, especially with increasements of the number of nodes. Motivated by the challenges caused by the stiffness, one remedy is to apply Krylov subspace methods proposed by Huang and his collaborators [43]. Instead of accepting the solution from simply fixed iteration, the optimal solutions are searched in the Krylov subspace. In detail, the Newton-Krylov method is applied to solve the nonlinear system

$$\widetilde{\mathbf{H}}(\widetilde{\mathbf{Y}}) = \mathbf{0}.$$

By KDC, the algorithm is guaranteed to convergence and the iteration is accelerated. Unfortunately, there are also drawbacks. More memory would be required and the overwork such as orthonormalization has to be calculated. The following example can be used to illustrate the improvement of KDC.

Example 2.2.2. We consider a simple problem

$$\begin{cases} \vec{y}'(t) &= \vec{p}'(t) - B(\vec{y}(t) - \vec{p}(t)), \\ \vec{y}(0) &= \vec{p}(0), \end{cases}$$

where $\vec{y}(t)$ and $\vec{p}(t)$ are vectors of dimension N. The exact solution is given by $\vec{y}(t) = \vec{p}(t)$. The matrix B is constructed by

$$B = U^T \Lambda U$$

where U is a randomly generated orthogonal matrix, and Λ is a diagonal matrix whose diagonal entries $\{\lambda_i\}_{i=1}^N$ are all positive. For $\vec{p}(t)$, we choose the ith component as $\cos(t + \alpha_i)$ with phase parameter $\alpha_i = \frac{2\pi}{N}$. We set the dimension of the system to 10, and use 10 Lobatto nodes in the simulation. We use $\Delta t = 0.1$ and study one time step. We set $\lambda_1 = \mathcal{O}(10^7)$ and the rest to the order of $\mathcal{O}(10^{-1})$. The result is plotted in Figure 2.2.

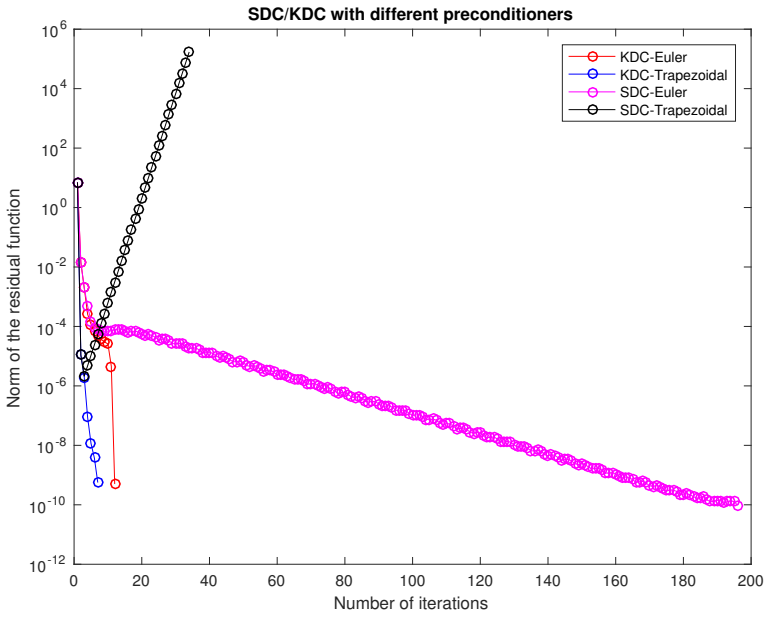


Figure 2.2: Deferred correction for the linear multimode problem

Note for SDC with Euler's preconditioning, it takes 196 iterations to converge. While a switch to Trapezoidal's preconditioning, the iteration diverges soon. With the acceleration of the Krylov

subspace, it takes 12 iterations to converge for Euler's preconditioning and 7 iterations for Trapezoidal's preconditioning. We can see that the convergence is significantly accelerated in this problem. In addition, although the Trapezoidal's rule is not stable for SDC, it converges efficiently for KDC.

2.3 Convergence analysis

In this section, we analyze the convergence of deferred correction for ODEs. The emphasize is on the stiff systems, in which order reduction is observed.

The standard reduction is used [91]. Start from the original problem $y'(t) = f(t, y)$ and let $y_0(t)$ denote a particular solution of the ODE we are interested in. If we make the substitution $y(t) = y_0(t) + u(t)$, then by linearization, we consider the problem $u'(t) = A(t)u(t)$. We then freeze the coefficient by setting $A = A(t_0)$ for some t_0 of interest. The idea here is that instability and stiffness are fundamentally transient phenomena, which may appear near some time t_0 and not others. The result is the constant coefficient linear problem. By diagonaliazation, we focus on the scalar problem $u'(t) = \lambda u(t)$.

Remark 2.3.1. This argument is not always true, Trefethen [91] discuss the potential failures of the approach. Nevertheless, it has achieved lots of success in applications.

In the scalar constant coefficient problem, we have the collocation formulation

$$\mathbf{Y} = \mathbf{Y}_0 + \Delta t \lambda S \mathbf{Y}.$$

By Picard's iteration, the solution is updated by Neumann series

$$\mathbf{Y}^{[n+1]} = C\mathbf{Y}^{[n]} + b,$$

where

$$C_{pic} = \Delta t \lambda S,$$
$$b_{pic} = \mathbf{Y}_0.$$

It is obvious that if λ is small, then we obtain a factor of $\Delta t \lambda$ in each update; on the other hand, if $\Delta t \lambda \gg 1$, then the iteration diverge. By some simple algebras, it turns out the deferred correction

methods for the problem is also equivalent to apply the Neumann series, but with preconditioning technique so that

$$C_{dc} = I - (I - \Delta t\lambda\widetilde{S})^{-1}(I - \Delta t\lambda S),$$
$$b_{dc} = (I - \Delta t\lambda\widetilde{S})^{-1}\mathbf{Y}_0.$$

Then it is sufficient to analyze the eigenstructure of the convergence matrix C if it is diagonalizable.

Theorem 2.3.1. *For linear ODE IVPs, the deferred correction iterations are convergent if and only if the spectral radius $\rho(C)$ (the supremum among the absolute values of all the eigenvalues) of the correction matrix C is less than 1.*

Motivated by the theorem, we define the "convergence region" to measure when the deferred correction methods are convergent for linear problems:

Definition 2.3.1. For linear ODE IVPs, we define the "convergence region" Ω of a deferred correction method as $\Omega = \{\lambda\Delta t : \rho(C(\lambda\Delta t)) < 1,\ \lambda \in \mathbb{C}\}$. The method is called "A-convergent" if Ω contains the left half complex plane. It is called "L-convergent" if it is "A-convergent" and $\lim_{|\lambda\Delta t|\to\infty} \rho(C(\lambda\Delta t)) \to 0$ for $\lambda\Delta t$ on the left half complex plane.

We find it very challenging to analyze in general scenario. To reduce the burden, we begin by considering asymptotic cases, then we numerically compute the contour lines of $\rho(C)$ to give an intuitive explanation.

2.3.1 Non-stiff systems

For the case λ is small, recall

$$C_{dc} = (I - \Delta t\lambda\widetilde{S})^{-1}\Delta t\lambda(\widetilde{S} - S).$$

We also obtain a factor of $\Delta t\lambda$ in each iteration. Furthermore, we expect the low frequencies errors are reduced better than Picard's iteration due to the term $\widetilde{S} - S$. The argument is verified in the following toy example.

Example 2.3.1. Consider the ODE

$$\begin{cases} (y(t) - \cos(t))' = \lambda(y(t) - \cos(t)), \\ y_0 = 1. \end{cases}$$

Apparently the analytical solution is $y(t) = \cos(t)$. This toy example can be used to understand the stiffness of the system and we will use it over and over again in the rest of the thesis. For non-stiff system, we set $T = 1, \lambda = 1$ with 10 Lobatto nodes. The final error is 1.08×10^{-11}. From the Figure 2.3, we can see that the SDC converges slightly faster than Picard's iteration, but their convergence rate is about the same.

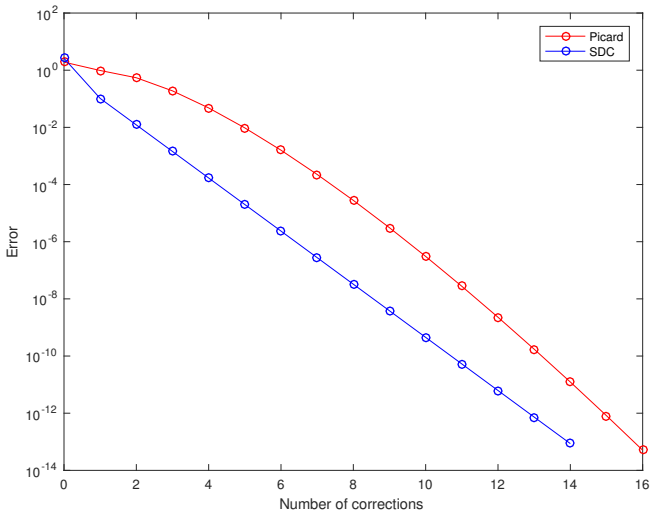


Figure 2.3: Comparison of deferred correction and Picard iteration

Remark 2.3.2. It is natural to wonder if the convergence can be accelerated by higher order preconditioners. It is shown in [40] if the nodes are chosen to be uniform, then in non-stiff system, the higher-order convergence is obtained by Runge-Kutta methods. We also obtain the similar results for the Trapezoidal's rule preconditioner.

2.3.2 Stiff systems

For the case when λ is large, the story is somewhat more complicated. Due to the stiffness, order reductions are observed. Recall that we obtain a factor of Δt in each iteration of deferred correction in non-stiff case, which is no longer true. To understand the convergence, if we consider

the backward Euler's method with Gauss nodes, in the strong stiff case such that $Re(\lambda) \to -\infty$, we want to understand the eigenstructure of the matrix

$$C_{dc} \sim I - \widetilde{S}^{-1} S.$$

With first glance, we notice that λ and Δt are canceled out, which explains the reduction of the order. The eigenvalues of the matrix are numerically solved to have a deeper insight. The result is summarized in Table 2.1.

n	2	3	4	5	6	7	8
	0.3170	0.4210	0.5610	0.6653	0.7420	0.7998	0.8448
n	9	10	11	12	13	14	15
	0.8805	0.9096	0.9337	0.9540	0.9713	0.9861	0.9991
n	16	17	18	19	20	25	50
	1.0105	1.0205	1.0295	1.0375	1.0448	1.0724	1.1280

Table 2.1: $\rho(I - \widetilde{S}^{-1} S)$ for different numbers of Gauss nodes, stiff case, *SDC.*

With increasements of the number of nodes, we find out the order reduction becomes more severe. In particular, in the strong stiff system, the SDC with more than 15 points will diverge. We have seen advantages in convergence by using higher-order preconditioners in the non-stiff cases, however, such successes are usually not observed in the stiff system. We analyze it in the same way and numerically calculate eigenvalues of the convergence matrix and show the results in the Table 2.2.

n	3	4	5	6	7	8	9
$\lvert\lambda\rvert_{max}$	0.3333	0.6180	0.8934	1.1658	1.4370	1.7076	1.9780
n	10	11	12	13	14	15	16
$\lvert\lambda\rvert_{max}$	2.2482	2.5183	2.7884	3.0585	3.3285	3.5986	3.8687
n	17	18	19	20	21	25	50
$\lvert\lambda\rvert_{max}$	4.1388	4.4089	4.6789	4.9490	5.2191	6.2995	13.0530

Table 2.2: $\rho(C)$ of SDC-Lobatto-T, strongly stiff limit case.

We observe that $|\lambda|_{\max}$ soon grows bigger than 1, with even 6 nodes, which explains the failure of the Trapezoidal's rule in the stiff case.

Other interesting observations in the past experiments are that by uniform nodes, the convergences are often better than the case with non-uniform nodes. By analyzing it, we have the following

theorem.

Theorem 2.3.2. *For the yp-formulation with uniform nodes, when $|\lambda\Delta t| \to \infty$, the correction matrix $\widetilde{S}^{-1}S - I$ has eigenvalues equal to zero; and its Jordan canonical form consists of one Jordan block.*

Remark 2.3.3. The uniform nodes discretization seem to have better convergence. However, it should be avoided with a large number of nodes due to the well-known Runge phenomenon.

Finally, if the left endpoint is included, then the simple asymptotic relation $C_{dc} \sim I - \widetilde{S}^{-1}S$ cannot be hold since the first row of $\widetilde{S}$ and S are zeros. In case, we re-write the matrix in the following form:

$$S = \begin{bmatrix} 0_{1\times 1} & \mathbf{0}_{1\times(m-1)} \\ S_{21} & S_{22} \end{bmatrix}$$

where $S_{21} \in \mathbb{R}^{(m-1)\times 1}$ and $S_{22} \in \mathbb{R}^{(m-1)\times(m-1)}$. Similarly, denote

$$\widetilde{S} = \begin{bmatrix} 0_{1\times 1} & \mathbf{0}_{1\times(m-1)} \\ \widetilde{S}_{21} & \widetilde{S}_{22} \end{bmatrix}.$$

Apply Woodbury matrix identity, we duduce that

$$C = \begin{bmatrix} 0 & \mathbf{0} \\ (I - \Delta t\lambda\widetilde{S}_{22}^{-1}\Delta t\lambda(S_{21} - \widetilde{S}_{21}) & I - (I - \Delta t\lambda\widetilde{S}_{22})^{-1}(I - \Delta t\lambda S_{22}) \end{bmatrix}.$$

Hence it is sufficiently to analyze the submatrix $I - (I - \Delta t\lambda\widetilde{S}_{22})^{-1}(I - \Delta t\lambda S_{22})$.

2.3.3 General cases

In general cases, we numerically compute the contour lines of the convergence of correction matrix in terms of different values of $\Delta t\lambda$. We focus on the Euler's method since the Trapezoidal's rule seem to be unstable in a stiff system.

We plot the contour with Gauss nodes in Figure 2.4. From the figure, we clearly see the convergence deteriorate with more points. Another thing we notice is that the convergence of deferred correction slows down near the purely oscillatory region. Such behaviors are also observed

when we calculate the dynamics of Schrödinger type equations. Similar results can be shown for different type quadratures. One interesting observation is that the yp-formulation with less than 5 uniform nodes is L-convergent [84].

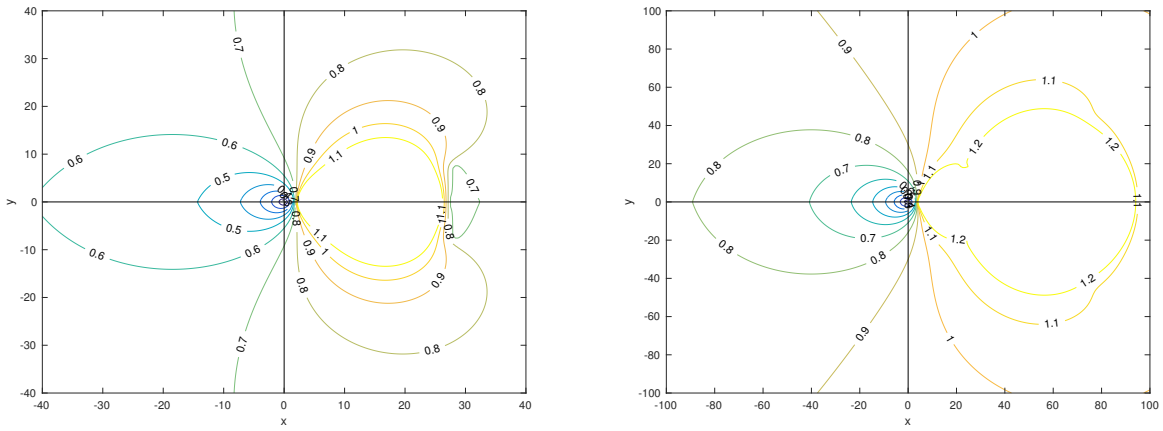


Figure 2.4: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 5$ and $m = 10$ for SDC, $\lambda = x + iy$

2.4 Diagonal preconditioners

In this section, we introduce a new class of diagonal preconditioners. The new preconditioner is very stable and easy to parallelize.

In order to achieve the local parallelization, we need to decouple the dependencies between the current node and the previous nodes. Hence, we update the error directly from the initial node. For stiff systems, with backward Euler's method, we have

$$\widetilde{\delta}(t_{n+1}) \approx t_{n+1}[\widetilde{f}(t_{n+1}, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_{n+1}, \widetilde{y})] + \epsilon(t_{n+1}).$$

For non-stiff systems, with forward Euler's method, we have

$$\widetilde{\delta}(t_{n+1}) \approx t_{n+1}[\widetilde{f}(t_0, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_0, \widetilde{y})] + \epsilon(t_{n+1}).$$

Since the error function is zero at the initial node, we realize this is actually the Picard's iteration. With this update, function evaluations at each node can be calculated simultaneously. In matrix

form, the backward Euler's integration matrix now becomes diagonal

$$\widetilde{S} = \begin{bmatrix} \Delta t_0 & 0 & 0 & \dots & 0 \\ 0 & \Delta t_1 & 0 & \dots & 0 \\ 0 & 0 & \Delta t_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \Delta t_{m-1} \end{bmatrix}.$$

Correspondingly, the preconditioner $I - \Delta t \widetilde{S}\mathbf{J}_{\mathbf{F}}(\widetilde{\mathbf{Y}})$ also becomes diagonal. For this reason, we refer this as the **diagonal preconditioner**. It is straightforward to generalize it to higher-order preconditioning techniques. For example, with Trapezoidal's rule, we have

$$\widetilde{\delta}(t_{n+1}) \approx \frac{t_{n+1}}{2}[(\widetilde{f}(t_0, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_0, \widetilde{y})) + (\widetilde{f}(t_{n+1}, \widetilde{y} + \widetilde{\delta}) - \widetilde{f}(t_{n+1}, \widetilde{y})] + \epsilon(t_{n+1}).$$

To accelerate the convergence, one could also adopt the KDC framework with diagonal preconditioners. To understand the behavior of this new class of preconditioners, we apply the similar analysis as in the last section. For non-stiff systems, the results are pretty similar. We focus on the stiff systems. For this particular type of preconditioners, in the strong stiff case, we are able to provide some analytical results. Assuming $\Delta t = 1$ and $\lambda = \frac{1}{\epsilon}$, where $\epsilon \to 0$ and we want to calculate the eigenvalues of the preconditioned system

$$\begin{aligned} A &= \left(I + \frac{1}{\epsilon}\widetilde{S}\right)^{-1}\left(I + \frac{1}{\epsilon}S\right) \\ &= (\epsilon I + \widetilde{S})^{-1}(\epsilon I + S). \end{aligned}$$

Assume μ is the eigenvalue of A, we deduce that

$$\begin{aligned} A\mathbf{F} &= \mu\mathbf{F} \\ \Rightarrow (\epsilon I + S)\mathbf{F} &= \mu(\epsilon I + \widetilde{S})\mathbf{F}. \end{aligned}$$

We take the asymptotic expansion with first two terms, assume that

$$\mathbf{F} \sim \mathbf{F}_0 + \epsilon \mathbf{F}_1,$$

$$\mu \sim \mu_0 + \epsilon \mu_1.$$

Substitute those into the equation,

$$(\epsilon I + \epsilon S)(\mathbf{F}_0 + \epsilon \mathbf{F}_1) = (\mu_0 + \epsilon \mu_1)(\epsilon I + \widetilde{S})(\mathbf{F}_0 + \epsilon \mathbf{F}_1). \tag{2.4.1}$$

By matching the $\mathcal{O}(1)$ terms,

$$S\mathbf{F}_0 = \mu_0 \widetilde{S} \mathbf{F}_0.$$

Matching the $\mathcal{O}(\epsilon)$ yields

$$\mathbf{F}_0 + S\mathbf{F}_1 = \mu_0 \widetilde{S} \mathbf{F}_1 + \mu_0 \mathbf{F}_0 + \mu_1 \widetilde{S} \mathbf{F}_0.$$

We are able to provide following theorems regarding the backward Euler's method and Trapezoidal's rule.

Theorem 2.4.1. *If $\widetilde{S}$ is discretized by backward Euler's method, then*

$$\begin{cases} [\mu_0]_n = \frac{1}{n} \\ [\mathbf{F}_0]_n = [t_1^{n-1}, \ldots, t_m^{n-1}]^T. \end{cases}$$

If endpoints are not included, then

$$\begin{cases} [\mu_1]_n = 0, \\ [\mathbf{F}_1]_n = -(n-1)^2 [t_1^{n-2}, \ldots, t_m^{n-2}]^T. \end{cases}$$

The proof is straightforward and we neglect the details. By the theorem, we show that the spectral radius of the diagonal preconditioner by backward Euler's method is bounded in the strong stiff system, which is not true for the triangular preconditioner. For more general case, we numerically

compute convergence region (contour=1) and other contour lines of $\rho(C)$ for triangular and diagonal preconditioners.

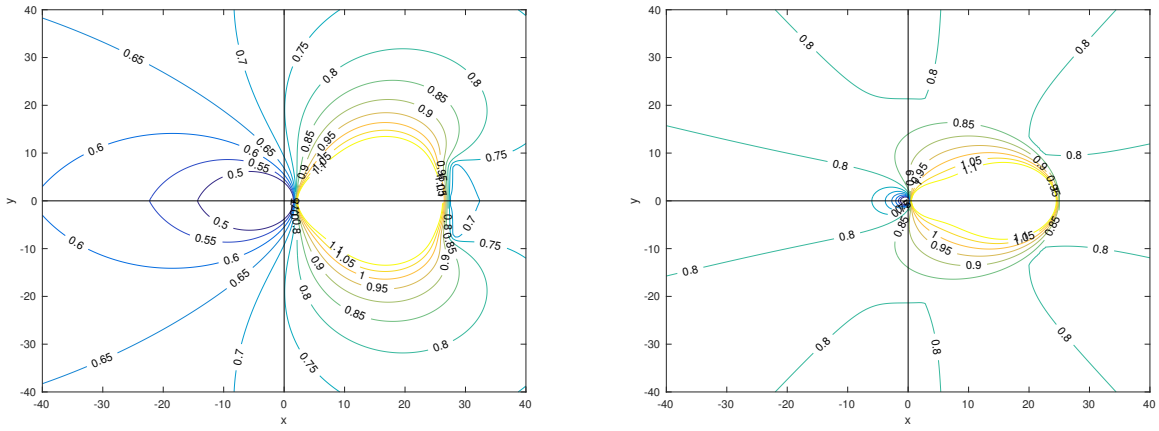


Figure 2.5: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 5$ for triangular (left) and diagonal (right) preconditioners, $\lambda = x + iy$

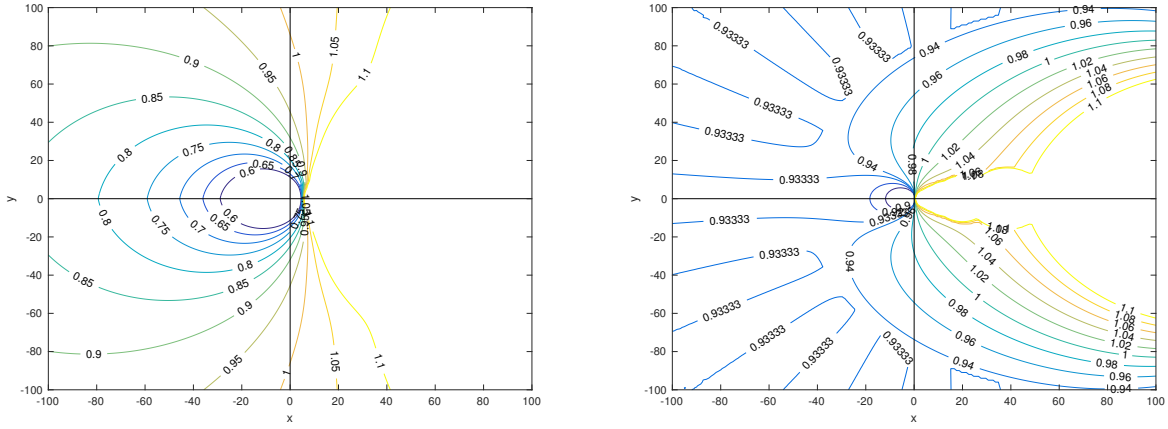


Figure 2.6: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 15$ for triangular (left) and diagonal (right) preconditioners, $\lambda = x + iy$

In Figures 2.5, we observe for 5 nodes, both preconditioners are A-convergent, and the triangular preconditioners have better behaviors. We then increase the number of Gauss nodes to 15 as plotted in Figure 2.6. The triangular preconditioner is no longer A-convergent, where the diagonal preconditioner still is. This indicates that the diagonal preconditioner is more stable than the triangular preconditioner, but the triangular preconditioner has better convergence rate in small to moderate λ regions.

Let's consider a special case, the region bounded by the contour of the $|\mu_0|_{max}$, that is, excluding the highly oscillatory data. In this case, if we require the error be decayed by a factor of ϵ, the number of function evaluations by diagonal preconditioners is approximate $-m^2 \log(\epsilon)$. The explicit form of eigenvalues by the triangular preconditioners are not known, but it can be verified numerically that the number of function evaluations is also large. For both preconditioners, the SDC is too expensive to apply directly. One remedy is to adopt the KDC, otherwise, practitioners recommend to reduce the step-size. However, if parallel computing is taken into account, then ideally computational complexity can be reduced by a factor of m by the diagonal preconditioners and is approximately equal to $\log(\epsilon)$ iterations of SDC by the triangular preconditioners. This provides us an alternative approach solving the stiff system with large time step effectively.

We then analyze the case of Trapezoidal's rule.

Theorem 2.4.2. *If $\widetilde{S}$ is discretized by Trapezoidal's rule with the left endpoint, for $n = 1$,*

$$\begin{cases} [\mu_0]_1 = 1, \\ [\mathbf{F}_0]_1 = [1, \ldots, 1]^T. \end{cases}$$

For $n > 1$, we have

$$\begin{cases} [\mu_0]_n = \frac{2}{n}, \\ [\mathbf{F}_0]_n = [t_1^{n-1}, \ldots, t_m^{n-1}]^T. \end{cases}$$

For $\mathcal{O}(\epsilon)$ with $n \geq 2$,

$$\begin{cases} [\mu_1]_n = 0, \\ [\mathbf{F}_1]_n = ((n-1) - (n-1)^2)[t_1^{n-2}, t_m^{n-2}]^T. \end{cases}$$

The spectral radius of the diagonal preconditioner by Trapezoidal's rule in the strong stiff case is also bounded, which is not held for triangular preconditioners. We also plot the contour lines in Figures 2.7 and 2.8.

It seems like the diagonal preconditioners with Trapezoidal's rule outperform the Euler's method in most regions and therefore is recommended. With the similar argument, for the case that

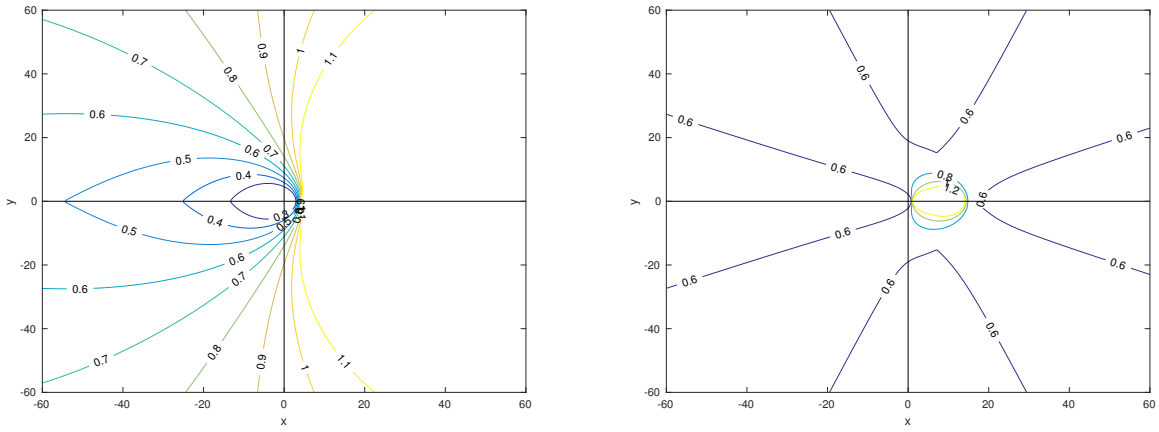


Figure 2.7: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 5$ for triangular (left) and diagonal (right) preconditioners, $\lambda = x + iy$

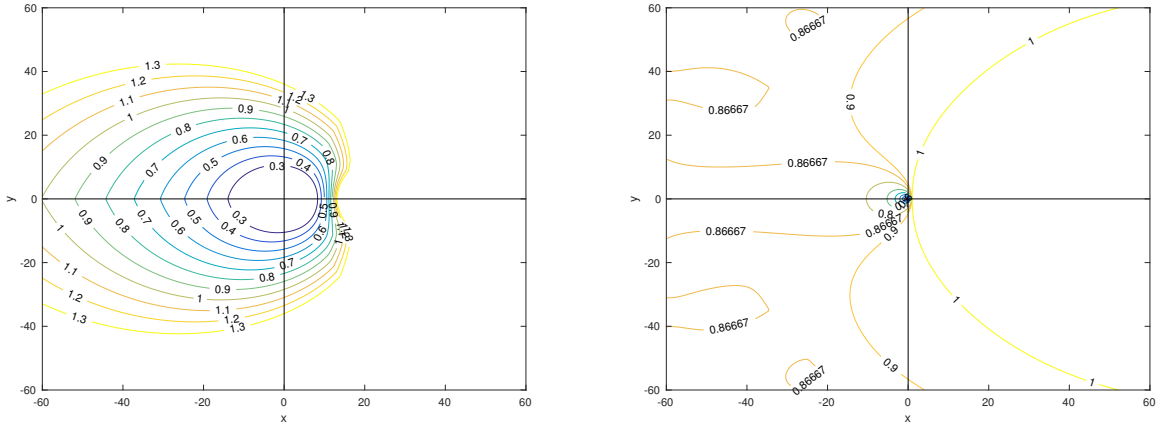


Figure 2.8: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 5$ for triangular (left) and diagonal (right) preconditioners, $\lambda = x + iy$

eigenvalues are bounded by the contour line of $|\mu_0|_{max}$, now the iteration number is reduced by a factor of 2 compared to the Euler's method.

We illustrate the performance of diagonal preconditioners with comparison with triangular preconditioners in the following example.

Example 2.4.1 (Linearized Richards' equation)**.** Consider the linearized Richards' equation,

$$M(x)u_t(x,t) = N(x)u_{xx}(x,t) + f(x,t),$$

where

$$M(x) = 2 + \cos(2\pi x),$$
$$N(x) = 2 + \cos(4\pi x),$$

and the analytical solution is given by

$$u(x,t) = e^{\cos(2\pi(x+t^2))-kt}.$$

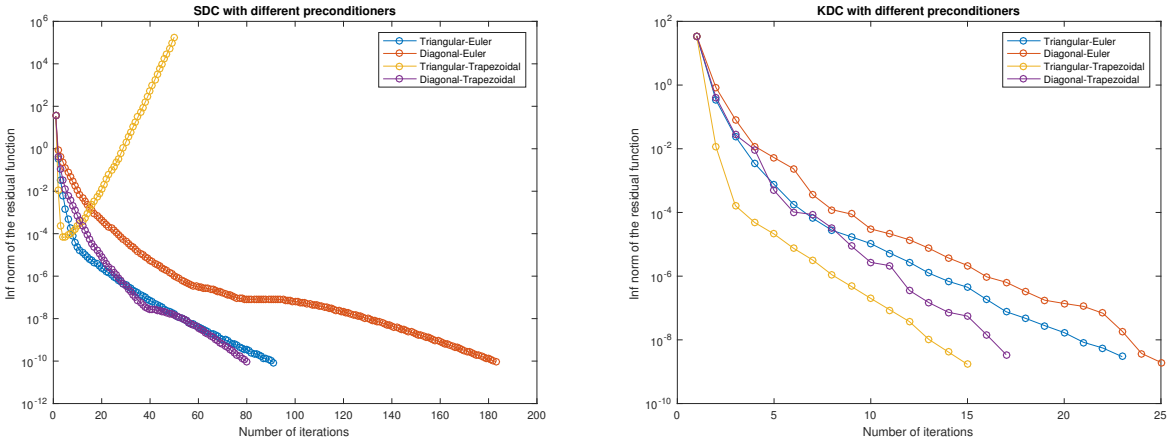


Figure 2.9: SDC (left) and KDC (right) for different preconditioners

The convergences of deferred correction with different preconditioners are plotted in Figure 2.9. First, we observe that triangular precondtioners converge faster than diagonal preconditioners in first few iterations since they have better convergence in non-stiff regions. Then the convergence slows

down, and eventually the diagonal preconditioner converges similarly to the triangular preconditioner with Euler's method. With the acceleration of the Krylov method, all preconditioners work well, but we could expect better performance for diagonal preconditioners in a parallel environment.

2.5 Optimal preconditioners

In this section, we introduce a new technique to improve the performance of the existing preconditioners without the usage of Krylov subspace.

Recall that in matrix language, the residual function at nth iteration is defined as

$$\boldsymbol{\epsilon}^{[n]} = \mathbf{Y}_0 + \Delta t S \mathbf{F}(\mathbf{Y}^{[n]}) - \mathbf{Y}^{[n]}.$$

One iteration of deferred correction yields that

$$\boldsymbol{\delta}^{[n+1]} = \Delta t \widetilde{S}[\mathbf{F}(\mathbf{Y}^{[n]} + \boldsymbol{\delta}^{[n+1]}) - \mathbf{F}(\mathbf{Y}^{[n]})] + \boldsymbol{\epsilon}^{[n]}.$$

Then the next step residual function is

$$\boldsymbol{\epsilon}^{[n+1]} = \Delta t S[\mathbf{F}(\mathbf{Y}^{[n]} + \boldsymbol{\delta}^{[n+1]}) - \mathbf{F}(\mathbf{Y}^{[n]})] + \boldsymbol{\epsilon}^{[n]}.$$

To obtain the accurate solution, we minimize the 2-norm of the residual function. The optimal solution is obviously using the spectral integration S itself. In order to make the preconditioner efficient, certain restrictions must be posted. With such requirement, we give the following definition.

Definition 2.5.1. Satisfying the certain restrictions, the "optimal" preconditioner is defined by minimizing the 2-norm of the residual function.

Now we discuss one particular example. Considering the stiffness comes from a linear operator $\mathcal{L}$. We want to solve the collocation formulation

$$(I - \Delta t \mathcal{L} S)\mathbf{Y} = \mathbf{Y}_0.$$

For the preconditioner, first-order and lower-triangular shape conditions are posed for efficiency and

accuracy considerations. In this case, the preconditioner matrix has the form of

$$\widetilde{S}(\alpha) = \begin{bmatrix} \alpha_{1,1} & 0 & 0 & \dots & 0 \\ \alpha_{2,1} & \alpha_{2,2} & 0 & \dots & 0 \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \alpha_{m,1} & \alpha_{m,2} & \alpha_{m,3} & \dots & \alpha_{m,m} \end{bmatrix}$$

with

$$\sum_i \alpha_{i,j} = t_i.$$

The question is how to choose the optimal choice of α. By some algebra, we deduce that the residual function is updated by

$$\boldsymbol{\epsilon}^{[n+1]} = (I - \Delta t \mathcal{L} S)(I - \Delta t \mathcal{L} \widetilde{S}(\alpha))^{-1} \boldsymbol{\epsilon}^{[n]}.$$

Hence by our definition, we require

$$\begin{cases} \min_\alpha \|(I - \Delta t \mathcal{L} S)(I - \Delta t \mathcal{L} \widetilde{S}(\alpha))^{-1} \epsilon^{[n]}\|_2^2, \\ \text{subject to first-order restriction.} \end{cases}$$

This is too difficult to solve. To simplify the calculation, we can get rid of the right-hand side so that the parameters can be precomputed and minimize the spectral radius of the convergence matrix,

$$\begin{cases} \min_\alpha \rho\left((I - \Delta t \mathcal{L} \widetilde{S}(\alpha))^{-1}(I - \Delta t \mathcal{L} S)\right), \\ \text{subject to first-order restriction.} \end{cases}$$

Since the optimization only needs to be calculated once, this can be done in precomputation stage.

We compare our new preconditioner with the triangular preconditioner in the Figure 2.10. We can see the improvement of the preconditioner in almost all regions. Moreover, in the strong stiff case when $\lambda \to -\infty$, the spectral radius is reduced from 0.92 to 0.3.

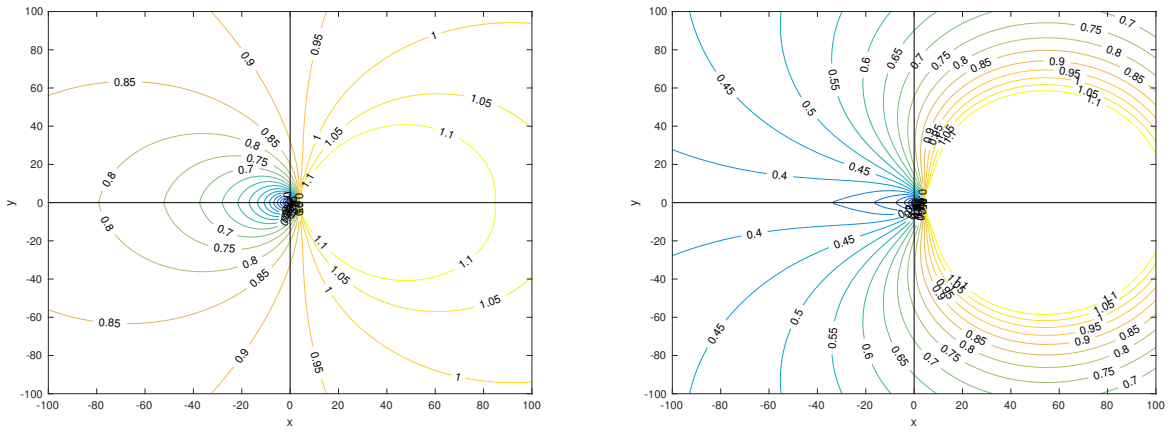


Figure 2.10: Contour lines of $\rho(C(\lambda\Delta t))$ for $m = 10$ for triangular (left) and "optimal" (right) preconditioners, $\lambda = x + iy$

The next example is provided to compare the performance of deferred correction with various preconditioners covered so far.

Example 2.5.1 (Diffusion equation with Dirichlet boundary condition)**.** Let's consider the Diffusion equation with Dirichlet boundary conditions:

$$u_t(x,t) = u_{xx}(x,t),$$
$$u_0(x) = \cos(x)$$

with the analytical solution:

$$u(x,t) = \cos(x)e^{-t}.$$

We use $N = 20$ Chebyshev points in the spatial direction and $p = 10$ Lobatto nodes in the temporal direction in one time step $T = 4$. This equation is very stiff that the largest eigenvalue of the Laplace operator grow as $\mathcal{O}(N^4)$. The performance of SDC and KDC are plotted following:

For SDC, since the equation is so stiff, we observe that the diagonal preconditioner with Trapezoidal's rule already outperforms the triangular preconditioner with Euler's method. Then we notice that our new preconditioner converges much faster than the rest of the preconditioners. But with the acceleration of the KDC, all preconditioners work pretty well.

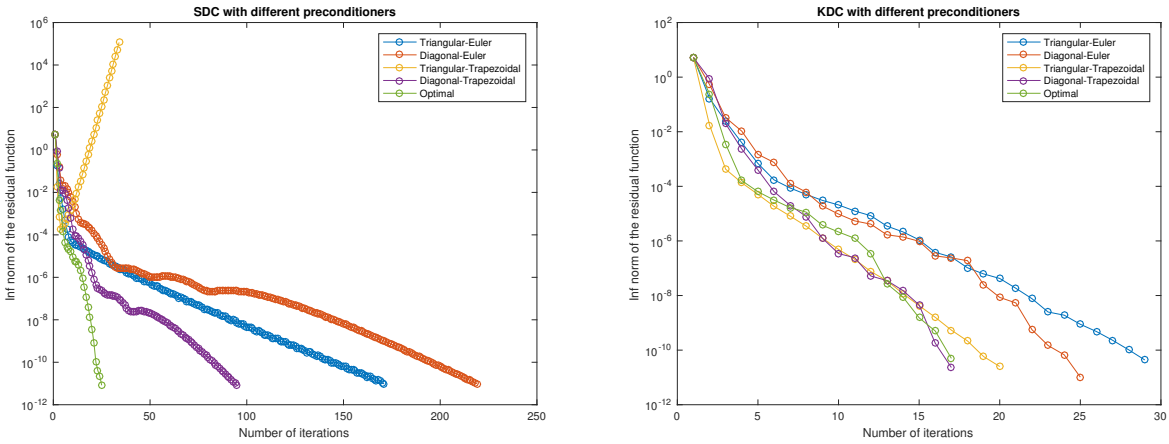


Figure 2.11: SDC (left) and KDC (right) for different preconditioners

2.6 Integral equation methods

In this section, we introduce a class of integral equation methods for stiff PDEs.

Consider the PDE of the form

$$\frac{\partial}{\partial t}u(x,t) = \mathcal{L}u(x,t) + \mathcal{N}(x,t,u),$$

where $\mathcal{L}$ stiff linear and $\mathcal{N}$ non-stiff nonlinear. In general, assume G is the Green's function for the operator $\frac{\partial}{\partial t} - \mathcal{L}$, then the solution can be written as

$$u(x,t) = \int_\Omega G(x-y,t)u_0(y)\ dy + \int_0^t \int_\Omega G(x-y,t-\tau)\mathcal{N}(y,\tau,u)\ dy\ d\tau. \tag{2.6.1}$$

For simplicity, we focus on the cases of free-space or periodic boundary conditions, in which the Fourier calculation yields that

$$u(x,t) = e^{\mathcal{L}t}u_0(x) + \int_0^t e^{-\mathcal{L}(\tau-t)}\mathcal{N}(x,\tau,u)\ d\tau. \tag{2.6.2}$$

Assuming the Fourier coefficient of $\mathcal{F}\{\mathcal{L}u\}$ is $g(k)$, then transforming in Fourier space gives us

$$\widehat{u}(k,t) = e^{g(k)t}\widehat{u}_0(k) + \int_0^t e^{-g(k)(\tau-t)}\mathcal{F}\{\mathcal{N}(x,\tau,u)\}\ d\tau. \tag{2.6.3}$$

Given the value of the solution $\widetilde{u}$, we can interpolate the nonlinear term $\widetilde{\mathcal{N}}(x,\tau,u)$, then the solution

can be recovered by

$$\mathcal{F}\{\widetilde{u}(x,t)\} = e^{g(k)t}\widehat{u}_0(k) + \int_0^t e^{-g(k)(\tau-t)}\mathcal{F}\{\widetilde{\mathcal{N}}(x,\tau,\widetilde{u})\}\ d\tau. \tag{2.6.4}$$

To uniquely determine the solution, Eq. (2.6.4) must be satisfied at each collocation points and we refer it as the collocation formulation for the integral equation method. For this new formulation of the discretization, the super-convergence can also be observed:

Theorem 2.6.1. *For the ODE IVP*

$$\begin{cases} y'(t) &= \lambda y(t) + f(t,y), \\ y(0) &= y_0. \end{cases}$$

If we apply the collocation formulation scheme to the integral equation

$$y(t) = e^{\lambda t}y_0 + \int_0^t e^{-\lambda(\tau-t)}f(\tau,y)\ d\tau$$

with p Gauss nodes, the solution is of order $2p$.

Proof. By some algebra, we show that

$$\left(e^{-\lambda t}\widetilde{y}(t)\right)' = e^{-\lambda t}\widetilde{f}(t,\widetilde{y}).$$

By collocation formulation, this equation must be satisfied at each collocation points, which impies that $\widetilde{f}(t,\widetilde{y}) + \lambda\widetilde{y}(t) - \widetilde{y}'(t)$ must have roots at each collocation points. On the other hand, we can show that

$$\epsilon(t) = \int_0^t e^{-\lambda(\tau-t)}[\widetilde{f}(\tau,\widetilde{y}) + \lambda\widetilde{y}(\tau) - \widetilde{y}'(\tau)]\ d\tau.$$

With the help of the theorem in [92], for $\lambda\Delta t \ll 1$, we obtain $2p$ order.

□

It is also interesting to see whether conservation laws are preserved. First, we give the definition of the first integral.

Definition 2.6.1. A non-constant function $I(y)$ is a first integral of $y'(t) = f(y)$ if

$$I'(y)f(y) = 0, \quad \forall y.$$

This is equivalent to the property that every solution $y(t)$ of $y'(t) = f(y)$ satisfies $I(y(t))$ =constant.

With some analysis, we show that the new formulation is no longer symplectic, but the conservation laws are preserved in some sense.

Theorem 2.6.2. *The quadratic first integral is preserved for* $\Delta t\lambda \ll 1$.

Proof. Let $\widetilde{y}(t)$ be the collocation polynomial of the discretization of the Gauss nodes, and assume that $I(y) = y^T Cy$, with symmetric C, is a first integral of $y'(t) = f(y)$. Since $\frac{d}{dt}I(\widetilde{y}(t)) = 2\widetilde{y}^T(t)C\widetilde{y}$,

$$\widetilde{y}^T(t_1)C\widetilde{y}(t_1) - \widetilde{y}^T(t_0)C\widetilde{y}(t_0) = 2\int_{t_0}^{t_1} \widetilde{y}(t)^T C\widetilde{y}'(t)\ dt.$$

From the proof of the last theorem, we know the function inside of the integral has the roots at each collocation points. Differentiate $\widetilde{y}(t)$ we obtain a combination of exponential function and polynomials, with the same theorem in the last proof, the error of the first integral is 0 provided $\Delta t\lambda \ll 1$. □

To solve the system efficiently, iterative methods are used. Since the stiffness is "removed", we apply explicit methods. The simplest scheme is the Picard's iteration

$$\mathcal{F}\{u^{[n+1]}(x,t)\} = e^{g(k)t}\widehat{u}_0(k) + \int_0^t e^{-g(k)(\tau-t)}\mathcal{F}\{\widetilde{\mathcal{N}}(x,\tau,u^{[n]})\}\ d\tau.$$

On the other hand, the deferred correction method can be used to reduce the number of iterations. The quality of the error is measured by the residual function

$$\widehat{\epsilon}(k,t) = e^{g(k)t}\widehat{u}_0(k) + \int_0^t e^{-g(k)(\tau-t)}\mathcal{F}\{\widetilde{N}(x,\tau,\widetilde{u})\}\ d\tau - \mathcal{F}\{\widetilde{u}(x,t)\}.$$

The error equation can be derived

$$\mathcal{F}\{\widetilde{\delta}(x,t)\} = \int_0^t e^{-g(k)(\tau-t)}\mathcal{F}\{\widetilde{N}(x,\tau,\widetilde{u}+\widetilde{\delta}) - \widetilde{N}(x,\tau,\widetilde{u})\}\ d\tau + \widehat{\epsilon}(k,t).$$

Locally, we have

$$\mathcal{F}\{\widetilde{\delta}(x,t_{n+1})\} = \mathcal{F}\{\widetilde{\delta}(x,t_n)\} + \int_{t_n}^{t_{n+1}} e^{-g(k)(\tau-t_{n+1})}\mathcal{F}\{\widetilde{\mathcal{N}}(x,\tau,\widetilde{u}+\widetilde{\delta}) - \widetilde{\mathcal{N}}(x,\tau,\widetilde{u})\}\, d\tau + \widehat{\epsilon}(k,t_{n+1}) - \widehat{\epsilon}(k,t_n).$$

Using the exponential time differencing forward Euler [93], the equation is approximated by

$$\begin{aligned}\mathcal{F}\{\widetilde{\delta}(x,t_{n+1})\} &\approx \mathcal{F}\{\widetilde{\delta}(x,t_n)\} + \int_{t_n}^{t_{n+1}} e^{-g(k)(\tau-t_{n+1})}\mathcal{F}\{\widetilde{N}(x,t_n,\widetilde{u}+\widetilde{\delta}) - \widetilde{\mathcal{N}}(x,t_n,\widetilde{u})\}\, d\tau + \widehat{\epsilon}(k,t_{n+1}) - \widehat{\epsilon}(k,t_n)\\ &= \mathcal{F}\{\widetilde{\delta}(x,t_n)\}\frac{e^{g(k)\Delta t_n}-1}{g(k)}\mathcal{F}\{\widetilde{N}(x,t_n,\widetilde{u}+\widetilde{\delta}) - \widetilde{N}(x,t_n,\widetilde{u})\} + \widehat{\epsilon}(k,t_{n+1}) - \widehat{\epsilon}(k,t_n).\end{aligned}$$

Remark 2.6.1. Analog to SDC, the integral with integrating factor can be precomputed.

Remark 2.6.2. It is also possible to correct the answer by higher-order methods in the class of exponential time differencing methods. However, we don't think the performance will be improved a lot since higher-order explicit methods require more computations in each calculation.

To understand the algorithm better, the matrix version is also presented. Discretize in time yields that

$$\mathbf{U} = e^{\mathcal{L}t}\mathbf{U}_0 + E\mathcal{N}(x,t,\mathbf{U}),$$

where E is the spectral integration matrix with integrating factor $e^{\mathcal{L}t}$. To solve the system, the Picard's iteration gives

$$\mathbf{U}^{[n+1]} = e^{\mathcal{L}t}\mathbf{U}_0 + E\mathcal{N}(x,t,\mathbf{U}^{[n]}).$$

We can also apply the more sophisticated triangular preconditioner by deferred correction scheme for better convergence. The residual function is calculated by

$$\boldsymbol{\epsilon} = e^{\mathcal{L}t}\mathbf{U}_0 + E\mathcal{N}(x,t,\widetilde{\mathbf{U}}) - \widetilde{\mathbf{U}}.$$

We still have the same definition of the error function. After some simple algebra, we derive the

error equation

$$\boldsymbol{\delta} = E(\mathcal{N}(x,t,\widetilde{\mathbf{U}}+\boldsymbol{\delta}) - \mathcal{N}(x,t,\widetilde{\mathbf{U}})) + \boldsymbol{\epsilon}. \tag{2.6.5}$$

By deferred correction scheme, in each iteration, we solve

$$\widetilde{\boldsymbol{\delta}} = \widetilde{E}(\mathcal{N}(x,t,\widetilde{\mathbf{U}}+\widetilde{\boldsymbol{\delta}}) - \mathcal{N}(x,t,\widetilde{\mathbf{U}})) + \boldsymbol{\epsilon}, \tag{2.6.6}$$

where $\widetilde{E}$ is the low-order integration matrix with integrating factor. By calculation, it is easy to show that the Jacobian of the Picard's iteration is

$$\frac{\partial\widetilde{\boldsymbol{\delta}}}{\partial\mathbf{U}} = -(I - E\mathbf{J}_{\mathcal{N}})$$

and the Jacobian of the deferred correction is

$$\frac{\partial\widetilde{\boldsymbol{\delta}}}{\partial\mathbf{U}} = -\left[I - (I - \widetilde{E}\mathbf{J}_{\mathcal{N}})^{-1}(\widetilde{E} - E)\mathbf{J}_{\mathcal{N}}\right].$$

So we would expect the convergence of the deferred correction will be slightly faster than Picard's iteration. However, Picard's iteration belongs to the class of diagonal preconditioners, the function evaluation can be done simultaneously. Assume the initial guess is computed by the sequential exponential time differencing forward Euler's method, p temporal nodes are used, and it takes Mth iterations to converge, then the parallel efficiency is

$$\frac{p(M+1)}{p+M}.$$

If the stiff part of the system is "removed" by the IEM preconditioner, then the Picard's iteration has better parallelization efficiency than SDC.

In the following examples, we compare the performance of IEM to some existing popular algorithms including Strang splitting and exponential time-diffencing Runge-Kutta method (ETDRK). The parallelization efficiency is also verified in the example.

Example 2.6.1 (Nonlinear Schrödinger equation)**.** Consider the nonlinear Schrödinger equation, in

particular, the cubic Schrödinger equation,

$$u_t(x,t) = iu_{xx}(x,t) + iq|u|^2(x,t)u(x,t)$$

with the period boundary condition. It has lots of important applications in the propagation of light in nonlinear optical fibers and planar waveguides and to Bose-Einstein condensates confined to highly anisotropic cigar-shaped traps.

Transforming to Fourier space gives

$$\widehat{u}_t(k,t) = -ik^2\widehat{u}(k,t) + iq\mathcal{F}\{|u|^2(x,t)u(x,t)\}.$$

Using the integrating factor gives us

$$\widehat{u}(k,t) = e^{-ik^2t}\widehat{u}(k,0) + iq\int_0^t e^{ik^2(\tau-t)}\mathcal{F}\{|u|^2(x,\tau)u(x,\tau)\}d\tau.$$

This corresponds to $g(k) = -ik^2$ and $\mathcal{N} = |u|^2u$ in Eq. (2.6.3).

We consider the traveling soliton solution:

$$u(x,t) = \frac{0.2}{q}e^{i(2x-3.9t)}\text{sech}(\sqrt{0.1}(x-4t)),$$

where $q = 0.1$. 512 Fourier nodes are used in $[-35\pi, 35\pi]$ to march to the time $T = 3$. The inf norm of the density is checked. We compare the performance of the IEM by deferred correction and Picard's iteration with ETDRK4 and Strang Splitting methods. The performances are plotted in Figure 2.12. All methods are comparable in the mediate accuracy region, but the IEM has much better performance in high accuracy region. Another thing we notice is that the computational effort of IEM by Picard's iteration is no much worse than the deferred correction.

We also monitor the growth of errors of the solution and first integrals. The result is plotted in Figure 2.13 and Figure 2.14. The SS and IEM both preserve the conservation laws reasonable well and perform better than ETDRK4. Furthermore, the error of the solution grows slowly by IEM.

As we discussed, because the Picard's iteration belongs to the class of diagonal preconditioners, we are able to calculate the function parallel. For comparison, we fix the same step-size and the

Figure 2.12: Comparison for different methods

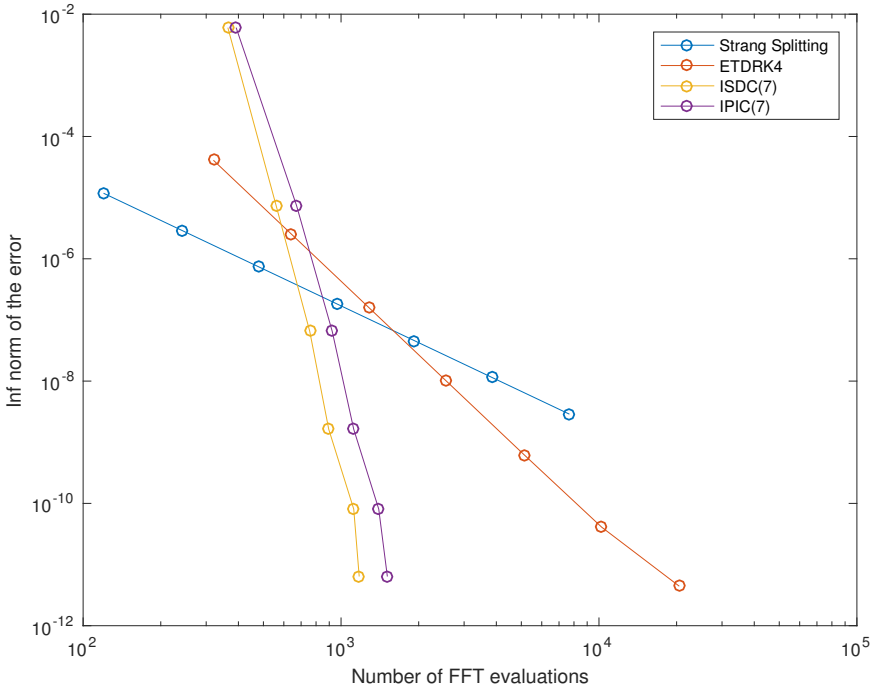


Figure 2.13: Conservation laws

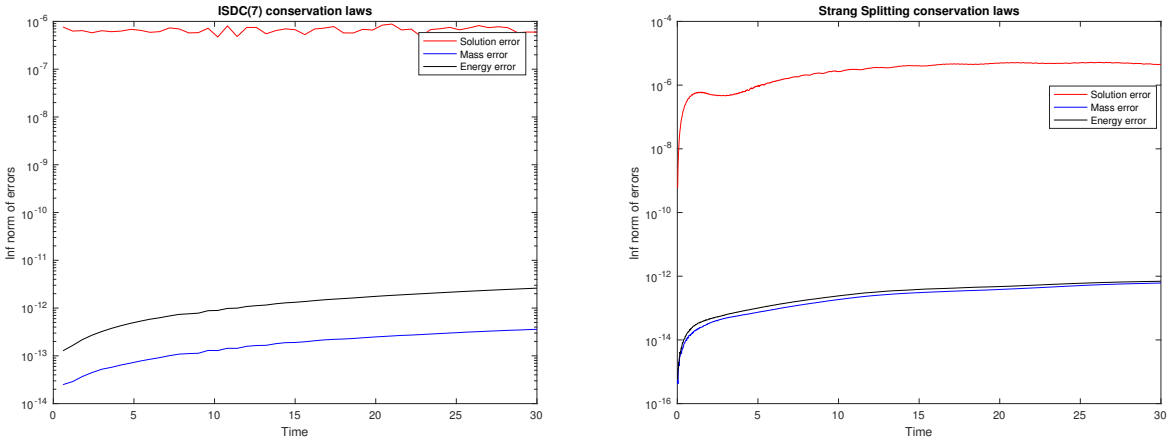
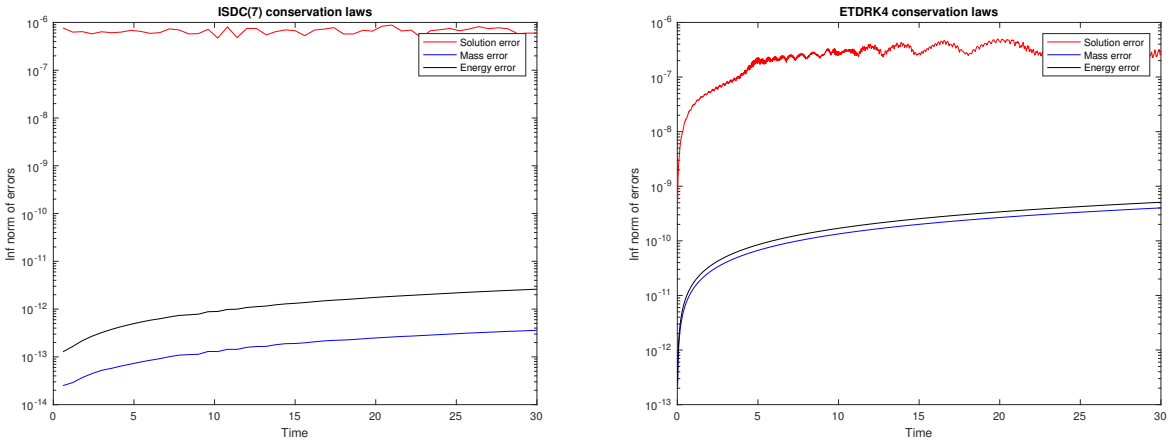


Figure 2.14: Conservation laws



tolerance. To mimic the two-dimensional case, we evaluate the Fourier transform N times in each calculation. We have the following result:

Method	Stepsize	Error	Iterations	Number of FFTs
IPIC(7)	0.2	2.44×10^{-12}	105	1.68×10^3
ISDC(7)	0.2	1.34×10^{-12}	75	1.26×10^3

The sequential Picard iteration takes around 12.92 seconds, whereas the parallel Picard iteration with 7 threads takes around 3.6 seconds. We obtain a speed up factor of 3.59, comparing to the theoretical speed factor 4. Thus, the IEM with Picard iteration is recommended in parallel environments.

Example 2.6.2 (Kuramoto-Sivashinksky equation)**.** Kuramoto-Sivashinksky equation is wided used in the study of a variety of reaction-diffusion systems. In one-dimensional case, the equation is written as

$$u_t(x,t) = -u(x,t)u_x(x,t) - u_{xx}(x,t) - u_{xxxx}(x,t)$$

with periodic boundary condition. The second-order term acts as an enegy source and has a destabilizing effect, and the nonlinear term transfers energy from low to high wavenumbers where the fourth-order term has a stabilizing effect. The equation is also very interesting from a dynamical

systems point of view, as it is a PDE that can exhibit chaotic solutions.

Transforming to Fourier spaces gives

$$\widehat{u}_t = -\frac{ik^2}{2}\widehat{u^2} + (k^2 - k^4)\widehat{u}.$$

Using the integrating factor, we have

$$\widehat{u}(k,t) = e^{(k^2-k^4)t}\widehat{u}(k,0) - \frac{ik^2}{2}\int_0^t e^{(k^4-k^2)(\tau-t)}\widehat{u^2}(k,\tau)\ d\tau.$$

This corresponds to $g(k) = k^2 - k^4$ and $\widehat{\mathcal{N}} = -\frac{ik^2}{2}\widehat{u^2}$ in Eq. (2.6.3).

We consider the initial condition

$$u(x,0) = \cos\left(\frac{x}{16}\right)\left(1 + \sin\left(\frac{x}{16}\right)\right).$$

The solution with this initial condition is extremely sensitive. The perturbations of the initial data will be amplified by as much as 10^8 up to $t = 150$. We simulate the dynamics with ETDRK4, integral equation method calculated by deferred correction and Picard's iteration and analytical inverting method.

Method	Stepsize	Error	Iterations	Number of FFTs
ETDRK4	3.75×10^{-3}	2.77×10^{-6}		3.2×10^5
ISDC(7)	1.875	4.87×10^{-6}	1544	3.90×10^4
IPIC(7)	1.875	4.84×10^{-6}	1958	4.90×10^4

Since the solution is chaotic, to have confidence in solutions, it is important to have a very accurate solution for such PDEs. Comparing to ETDRK4, the high-order methods have much better performance. The computational complexity is reduced by a factor of approximately $6-8$. If parallel computing is taken into account, we would expect even better performance. We believe our methods would have great advantages to simulate the chaotic systems.

2.7 Applications in time-dependent density functional theory

In section, we briefly discuss how the IEM can be applied to TDDFT.

Consider the TDKS equation of the form

$$i\frac{\partial}{\partial t}\phi_i(x,t) = \left[-\frac{\nabla^2}{2} + V_{KS}(x,t,\rho)\right]\phi_i(x,t), \tag{2.7.1}$$

and

$$\rho(x,t) = \sum_i^{occ} |\phi_i(x,t)|^2,$$

where

$$V_{KS}(x,t,\rho) = V_{ext}(x,t) + V_{Hartree}(x,\rho) + V_{xc}(x,t,\rho)$$

and

$$V_{Hartree}(x,\rho) = \int \frac{\rho(y,t)}{|x-y|}\, dy.$$

V_{xc} refers to the exchange correction potential. The analytical of the exchange correction potential is not known, there are several approaches can be used to approximate it and we will neglect the detail here.

We want to apply the IEM to the Eq. (2.7.1). The stiffness of the equation is mainly coming from the imaginary Laplace operator. The Hartree potential is a convolution, and one can prove that the convolution of this kernel is compact. The contribution of the exchange correction potential is deemed as a small quantity, and hence we believe it is non-stiff. The analytical form of the potential between the electron and nuclei in the external field can be very stiff. However, in the practice, pseudo-potential is usually used for approximation and then the term becomes non-stiff. By Fourier transform, we have

$$\frac{\partial}{\partial t}\widehat{\phi}_i(k,t) = -\frac{ik^2}{2}\widehat{\phi}_i(k,t) - i\mathcal{F}\{V_{KS}(x,t,\rho)\phi_i(x,t)\}.$$

By integrating factor, we deduce that

$$\widehat{\phi}_i(k,t) = e^{-\frac{ik^2}{2}t}\widehat{\phi}_i(k,0) - i\int_0^t e^{\frac{ik^2}{2}(\tau-t)}\mathcal{F}\{V_{KS}(x,\tau,\rho)\phi_k(x,\tau)\}\ d\tau.$$

With the reformulated integral equation representation, the stiffness is, therefore "removed". Then we are safe to apply the Picard's iteration with parallelization to update the solution

$$\widehat{\phi}_i^{[n+1]}(k,t) = e^{-\frac{ik^2}{2}t}\widehat{\phi}_i(k,0) - i\int_0^t e^{\frac{ik^2}{2}(\tau-t)}\mathcal{F}\{V_{KS}(x,\tau,\rho^{[n]})\phi_k^{[n]}(x,\tau)\}\ d\tau.$$

We have successfully implemented the IEM to the KSSOLV package [94]. However, the package is currently aimed to the small to moderate scale calculation, hence illustrating numerical examples are not available at this moment.

CHAPTER 3

Hierarchical Methods[1]

3.1 Introduction

Hierarchical methods are proven to be as one of the most successful tools in the spatial direction and have provided a rich history of mechanistic insights on physics, chemistry, material science, data science and so on. More specifically, we list some famous applications by hierarchical methods. (a) In the FMM[57, 95, 96], the information is compressed when target and source boxes are well-separated. Then the information is translated between parents and children and converted for the boxes in the interaction list recursively in the tree structure; (b) In the multi-level models in statistics (also referred to as hierarchical linear models or nested data models [97, 98]), data are often organized at more than one levels, and simple structures are often assumed for the models at each level and how different levels are connected; (c) Recently, in the convolutional neural network models in deep learning [99, 100, 101], learnable filters or kernels are introduced at different convolutional layers. The local connectivity is often assumed so the filters can compress the data by statistically fitting a set of parameters to match existing results from the learning database, and the compressed data are transmitted to parent layers as new inputs.

The key to the success of the hierarchical methods can be roughly summarized as follows:

Hierarchical tree structure: In the classical computational physics, a hierarchical tree structure is utilized by dividing the computational domain recursively to allow the efficient translation of the information. As a result, a computational effort can be reduced significantly. If the tree is rather uniform, then the number of tree nodes is approximate $\mathcal{O}(N)$ and the depth of the tree is normally $\mathcal{O}(\log N)$. If each level only requires $\mathcal{O}(N)$ operators, such as fast Fourier transform (FFT), the

[1]Some materials in this chapter previously appeared as an article in the arXiv and is already submitted. The original citation is as follows: M. Cho, J. Huang, D. Chen, and W. Cai. "A heterogeneous FMM for 2-D layered media Helmholtz equation I: two & three cases", arXiv preprint arXiv:1703.09136 (2017). Some materials in this chapter will appear in the future and are currently in preparation.

algorithm complexity will be $\mathcal{O}(N \log N)$; if each tree node only requires $\mathcal{O}(1)$ operators, such as FMM and multigrid method, an asymptotically optimal $\mathcal{O}(N)$ overall computational complexity can be obtained. In the statistics and data science, a hierarchical structure provides an accurate and efficient representation of the model.

Compression of data: Low-rank and low-dimensional features commonly exist in nature. In the example of particle systems, smoothness of the far-field expansion give us the low-rank representation of the potential function. It is extremely difficult to find a universal way to compress the data and each case should be dealt in the light of specific conditions.

Translation of data: The "sparsity" of the translation in the hierarchical structure avoid the direct interactions with all nodes. In the hierarchical structure, the compressed information can be translated very efficiently between each level.

Recursive Implementation and Parallelization: In many cases, algorithms can be implemented recursively instead of for- loop. This implementation commonly exists in the computational physics and provide not only an abstract yet simple approach but also a different perspective to understand the model. Furthermore, recursive algorithms can be easily interfaced with state-of-the-art dynamical scheduler (e.g., Cilk++ [102, 103, 104]) from High-Performance Computing community to allow the efficient parallelization in the modern computer architecture.

We are now armed with the necessary tools to apply an efficient hierarchical method for the spatial integral equation, starting with the fast solver for the two-point boundary value problem proposed in [32] as an example. By recasting the differential equation into the well-conditioned Fredholm integral equation of the second kind, the information of boundary conditions in the local problems is converted to the coupling coefficients, which is then translated between parents and children through the tree structure recursively. Furthermore, a parallel version of the algorithm is implemented based on Cilk++ multithreaded runtime system. The solver has been used a fundamental building block for the efficient solutions of time-dependent differential equations and can be applied to problems in higher dimensions when coupled with the spatial operator splitting techniques. In addition, the solver can be adapted as a preconditioner for SDC and KDC [42] as discussed in last Chapter.

We then apply our techniques to multi layered media problem in the acoustic and electromagnetic applications. The difficulty arises in the computation of the domain Green's function with integral representation. There have been many efforts to discretize the integral with less number of nodes

and then apply the fast algorithm for summation [60, 58, 59], which is considerably expensive in that the target integral is too complicate to compress. Alternatively, we observe that it is sufficient only to compress the free-space Green's function, and a similar result exists for the translation operators for the multipole and local expansions. All the spatially variant information of the domain Green's function are collected into the "multipole to local" translations and therefore the FMM becomes "heterogenous". One striking feature of our method is its efficiency: the computational complexity is similar to the evaluation of Green's function in free space and the algorithm can be orders of magnitude faster than existing schemes for simulating 2-D waves in two-layered media.

This chapter is organized as follows. Section 3.2 describes the parallel version of the fast solver for two-point boundary value problems. Section 3.3 introduces the heterogeneous FMM for layered media problem. The paper reporting on the results in Section 3.2 is in preparation [105] and the materials in Section 3.3 has appeared in [106].

3.2 Fast two-point boundary balue problem solver

In this section, we describe the fast recursive algorithm for the two-point boundary value problem [32] and how parallelization can be achieved.

Consider the general linear variable coefficient ordinary differential equations of the form

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x) \tag{3.2.1}$$

on an interval $[a, c] \subset \mathbb{R}$, where $p(x)$ and $q(x)$ are continuous function on (a, c), $u \in C^2[a, c]$ satisfies the linear boundary conditions

$$\begin{cases} \xi_{l0} \cdot u(a) + \xi_{l1} \cdot u'(a) = \Gamma_l, \\ \xi_{r0} \cdot u(c) + \xi_{r1} \cdot u'(c) = \Gamma_r \end{cases}$$

and $\xi_{l0}, \xi_{l1}, \xi_{r0}, \xi_{r1}, \Gamma_l$ and Γ_r are given constants.

3.2.1 Integral formulation and tree structure

To efficient represent the solution, the adaptive tree structure is employed to divide the interval into small subintervals as in Figure 3.1. It is not realistic to solve the whole problem directly; on the other hand, the local problem can be easily solved. To make the algorithm efficiently, it is desirable

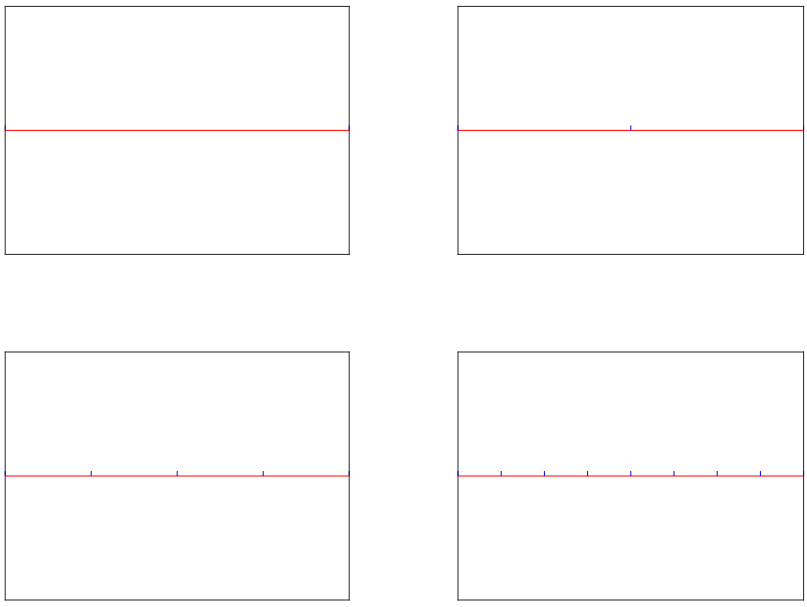


Figure 3.1: Binary tree for 1-D interval

to construct the global solution based on the local solutions and translate the information into the tree structure. The observation is that for two-point boundary value problem, if the boundary conditions at the endpoints are known, then the local solutions can be computed by only solving the local equation. However, it is not obvious how to derive the relationship of boundary conditions between the local problems and global problems. Alternatively, the integral equation formulation for its two major advantages:

- The convolution of Green's function contains global information and the information can be easily translated in the tree structure, making the algorithm more efficiently.
- Numerical integration is more stable than the numerical differentiation.

In details, the solution to the equation can be decomposed into two parts $u = u_i + u_h$ using the superposition principle, where u_i is a linear function satisfying the inhomogeneous boundary conditions and u_h solves the linear equation

$$u_h''(x) + p(x)u_h'(x) + q(x)u_h(x) = \widetilde{f}(x) \tag{3.2.2}$$

where

$$\widetilde{f}(x) := f(x) - (u_i''(x) + p(x)u_i'(x) + q(x)u_i(x))$$

with homogeneous boundary conditions

$$\begin{cases} \xi_{l0} \cdot u_h(a) + \xi_{l1} \cdot u_h'(a) & = 0, \\ \xi_{r0} \cdot u_h(c) + \xi_{r1} \cdot u_h'(c) & = 0. \end{cases}$$

We would like to represent the solution in terms of Green's function by making use of the following lemma [107].

Lemma 3.2.1. *Let $q_0 \in C^1(a, c)$ and suppose that the equation*

$$\phi''(x) + q_0(x)\phi(x) = 0 \tag{3.2.3}$$

subject to the homogeneous boundary condition has only the trivial solution. Then there exist two linearly independent functions $g_l(x), g_r(x)$ which satisfy Eq. (3.2.3) and the homogeneous boundary condition. Green's function for this equation, denoted by G_0, can be constructed as follows:

$$G_0(x,t) = \begin{cases} g_l(x)g_r(t)/s, & \text{if } x \le t, \\ g_l(t)g_r(x)/s, & \text{if } x \ge t, \end{cases}$$

where s is a constant given by

$$s = g_l(x)g_r'(x) - g_l'(x)g_r(x).$$

Given Green's function G_0, we take the ansaze that

$$u_h(x) = \int_a^c G_0(x,t) \cdot \sigma(t)\ dt, \tag{3.2.4}$$

where $\sigma(x)$ is the new unknown density function. Substitute Eqs. (3.2.4) into (3.2.2), we deduce

that

$$\sigma(x) + p(x)\int_a^c G_1(x,t)\sigma(t)\ dt + (q(x) - q_0(x))\int_a^c G_0(x,t)\sigma(t)\ dt = \widetilde{f}(x), \tag{3.2.5}$$

where

$$G_1(x,t) = \frac{d}{dx}G_0(x,t).$$

To obtain the explicit form of the Green's function, the following lemma from [32] is provided for some particular choice of $q_0(x)$.

Lemma 3.2.2. *If $|\xi_{l0}| \geq |\xi_{l1}|$ or $|\xi_{r0}| \geq |\xi_{r1}|$, then Green's function corresponding to $q_0(x) = 0$ can be constructed from*

$$g_l(x) = \xi_{l0}(x-a) - \xi_{l1},$$
$$g_r(x) = \xi_{r0}(x-c) - \xi_{r1}.$$

If both $|\xi_{l0}| < |\xi_{l1}|$ and $|\xi_{r0}| < |\xi_{r1}|$, then Green's function corresponding to $q_0(x) = -1$ can be constructed from

$$g_l(x) = \xi_{l1}cosh(x-a) - \xi_{l0}sinh(x-a),$$
$$g_r(x) = \xi_{r1}cosh(x-c) - \xi_{r0}sinh(x-c).$$

We define the operator $P : L^2[a,c] \to L^2[a,c]$ as follows:

$$P\eta(x) := \eta(x) + p(x)\int_a^c G_1(x,t)\eta(t)\ dt + (q(x) - q_0(x))\int_a^c G_0(x,t)\eta(t)\ dt.$$

Then Eq. (3.2.5) can be written compactly as

$$P\sigma = \widetilde{f}.$$

Since we have the explicit form of the Green's function, the equation can be simplified that

$$P\eta(x) = \eta(x) + \psi_l(x)\int_a^x g_l(t)\eta(t)\ dt + \psi_r(x)\int_x^c g_r(t)\eta(t)\ dt,$$

where

$$\psi_l(x) = \frac{p(x)g_r'(x) + (q(x) - q_0(x))g_r(x)}{s},$$
$$\psi_r(x) = \frac{p(x)g_l'(x) + (q(x) - q_0(x))g_l(x))}{s}.$$

3.2.2 Compression of data

With the help of the Green's function, the global information is encoded in the convolution. Let's restrict our attention to a local subinterval $B = [b_l, b_r] \subset [a, c]$, then for $x \in B$ we observe that

$$\begin{aligned} P\eta(x) &= \eta(x) + \psi_l(x)\int_{[a,b_l]\cup[b_l,x]} g_l(t)\eta(t)\ dt + \psi_r(x)\int_{[x,b_r]\cup[b_r,c]} g_r(t)\eta(t)\ dt \\ &= \eta(x) + \psi_l(x)\int_{b_l}^x g_l(t)\eta(t)\ dt + \psi_r(x)\int_x^{b_r} g_r(t)\eta(t)\ dt + \psi_l(x)\int_a^{b_l} g_l(t)\eta(t)\ dt + \psi_r(x)\int_{b_r}^c g_r(t)\eta(t)\ dt \\ &= P_B\sigma_B(x) - \psi_l(x)\lambda_l^B - \psi_r(x)\lambda_r^B \end{aligned}$$

where

$$P_B\eta(x) = \eta_B(x) + \psi_l(x)\int_{b_l}^x g_l(t)\eta_B(t)\ dt + \psi_r(x)\int_x^{b_r} g_r(t)\eta_B(t)\ dt \tag{3.2.6}$$

and

$$\lambda_l^B = -\int_a^{b_l} g_l(t)\sigma(t)\ dt,$$
$$\lambda_r^B = -\int_{b_r}^c g_r(t)\sigma(t)\ dt.$$

Now for $x \in B$, the solution can be written as

$$P\sigma(x) = P_B\sigma_B(x) - \psi_l(x)\lambda_l^B - \psi_r(x)\lambda_r^B = \widetilde{f}(x),$$

where

$$\lambda_l^B = -\int_a^{b_l} g_l(t)\sigma(t)\ dt,$$
$$\lambda_r^B = -\int_{b_r}^{c} g_r(t)\sigma(t)\ dt.$$

Definition 3.2.1. The constants λ_l^B and λ_r^B will be referred to as coupling coefficients.

If coupling coefficients are known, then the solution in B can be obtained by only solving the local discretization,

$$\sigma_B(x) = P_B^{-1}\widetilde{f}(x) + \lambda_l^B P_B^{-1}\psi_l(x) + \lambda_r^B P_B^{-1}\psi_r(x), \tag{3.2.7}$$

The coupling coefficients play important roles as boundary conditions in differential equations, which are compressed in this formulation.

3.2.3 Translation of data

To translate the information, the relationship of coupling coefficients between parents and children are derived analytically.

Parent to children At the moment, suppose that we already solves the η_B such that it satisfies

$$P_B\eta_B = \mu_l^B\psi_l + \mu_r^B\psi_r + \mu^B\widetilde{f}. \tag{3.2.8}$$

In parsBVP, B is subdivided into a left and a right subterinterval, denotes by D and E, respectively, and refer to D and E as B's children. Then the density function in D and E must satisfy

$$P_D\eta_D = \mu_l^D\psi_l + \mu_r^D\psi_r + \mu^D\widetilde{f},$$
$$P_E\eta_E = \mu_l^E\psi_l + \mu_r^E\psi_r + \mu^E\widetilde{f}.$$

To proceed further and simplify the notation, the following definitions are given.

Definition 3.2.2. Let X denote a subinterval of $[a, c]$. Then

$$\alpha_l^X \equiv \int_X g_l(t) P_X^{-1} \psi_l(t)\ dt, \quad \alpha_r^X \equiv \int_X g_r(t) P_X^{-1} \psi_l(t)\ dt,$$
$$\beta_l^X \equiv \int_X g_l(t) P_X^{-1} \psi_r(t)\ dt, \quad \beta_r^X \equiv \int_X g_r(t) P_X^{-1} \psi_r(t)\ dt,$$
$$\delta_l^X \equiv \int_X g_l(t) P_X^{-1} \widetilde{f}(t)\ dt, \quad \delta_r^X \equiv \int_X g_r(t) P_X^{-1} \widetilde{f}(t)\ dt.$$

The information of children can be derived from the information of parent by the next lemma from [32].

Lemma 3.2.3. *Suppose that we are given the coefficients μ_l^B, μ_r^B, μ^B in Eq.* (3.2.8)*. Then their refinements are given by*

$$\mu^D = \mu^B,$$
$$\mu^E = \mu^B,$$
$$\mu_l^D = \mu_l^B,$$
$$\mu_r^E = \mu_r^B,$$

and

$$\begin{bmatrix} \mu_r^D \\ \mu_l^E \end{bmatrix} = \begin{bmatrix} 1 & \alpha_r^E \\ \beta_l^D & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mu_r^B(1-\beta_r^E) - \mu^B \delta_r^E \\ \mu_l^B(1-\alpha_l^D) - \mu^B \delta_l^D \end{bmatrix}.$$

Based on the lemma, we observe that if we know the information from the parent, and if we solve the local system, then the information of children can be determined. If the system of children is still too large to solve, we will then continue to subdivide the children until locally the solution can be represented efficiently.

Children to parent In the finest level, the local information of α, β, δ can be solved efficiently. Then the following lemma from [32] help us to spread the information from the children to its parent.

Lemma 3.2.4. *Suppose that B is a subinterval with children D and E. Then*

$$\begin{aligned}
\alpha_l^B &= \frac{(1-\alpha_l^D)\cdot(\alpha_l^E - \alpha_r^E\beta_l^D)}{\Delta} + \alpha_l^D, \\
\alpha_r^B &= \frac{\alpha_r^E\cdot(1-\beta_r^D)\cdot(1-\alpha_l^D)}{\Delta} + \alpha_r^D. \\
\beta_l^B &= \frac{\beta_l^D\cdot(1-\beta_r^E)\cdot(1-\alpha_l^E)}{\Delta} + \beta_l^E, \\
\beta_r^B &= \frac{(1-\beta_r^E)\cdot(\beta_r^D - \beta_l^D\alpha_r^E)}{\Delta} + \beta_r^E. \\
\delta_l^B &= \frac{1-\alpha_l^E}{\Delta}\cdot\delta_l^D + \delta_l^E + \frac{(\alpha_l^E-1)\cdot\beta_l^D}{\Delta}\cdot\delta_r^E, \\
\delta_r^B &= \frac{1-\beta_r^D}{\Delta}\cdot\delta_r^E + \delta_r^D + \frac{(\beta_r^D-1)\cdot\alpha_r^E}{\Delta}\cdot\delta_l^D,
\end{aligned}$$

with $\Delta = 1 - \alpha_r^E\beta_l^D$.

Hence start from the finest level, the information can be translated recursively from children to their parents via the lemma efficiently.

3.2.4 Algorithm

With the technologies of compression and translation of data, a divide and conquer technique can be employed to efficiently solve the system.

1. **Generate tree.**

   Starting from the root interval $[a, c]$, recursively subdivide each interval into two smaller intervals until the intervals at the finest level are sufficiently small that the function can be well approxiamted by Chebyshev expansion. This procedure generates a binary tree with each node corresponding to a subinterval. The finest level intervals will be referred to as leaf nodes, others will be referred to as interval nodes.

2. **Solve local problem.**

   In the finest level, for each leaf node B_i, compute $P_{B_i}^{-1}\psi_l, P_{B_i}^{-1}\psi_r$, and $P_{B_i}^{-1}\widetilde{f}$ and then compute α, β, and δ correspondingly.

3. **Upward pass.**

   Translate the information from children to their parents by Lemma 3.2.4.

4. **Downward pass.**

Translate the information from parents to children by Lemma 3.2.3.

5. **Construct solution to integral equation.**

   Evaluate σ on each leaf node by Eq. (3.2.7).

6. **Construct solution to boundary value problem.**

   The solution can be constructed then easily.

3.2.5 Adaptive algorithm

The algorithm can be performed adaptively. We leave the detail of discussion in the original paper [32]. The key idea is to only subdivide the intervals when necessarily. The rule of thumb is the following.

- On each subinterval B_i, compute the monitor function

$$S_i = |\sigma_i^{N-2}| + |\sigma_i^{N-1} - \sigma_i^{N-3}|$$

  where σ_i is the Chebyshev coefficient of the density function

- Compute $S_{div} = \max_{i=1}^{M} S_i/2^C$, where $C > 1$ is provided by the user (recommended $C = 4$).

- If B_i and B_{i+1} are children of the same node and $(S_i + S_{i+1}) < S_{div}/2^N$, then replace them by their parent. This step merges subintervals which are determined to be overresolved.

Let u_r be the approximation to $u(x)$ at refinement stage r, the termination condition is setted by

$$\frac{\|u_r - u_{r-1}\|}{\|u_r + u_{r-1}\|} < tol.$$

3.2.6 Parallelization

The Cilk multithreaded runtime system extends the C/C++ language with only a handful of new keywords including *cilk_for, cilk_spawn*, and *cilk_sync*, and it automatically manages lower level aspects of parallelization including protocols, load balancing, scheduling, and other runtime issues. In particular, the Cilk scheduler is very effectively for recursive algorithms such as the upward and downward passes in the divide-and-conquer strategy in our implementation, which

can be mathematically described as recursive data processing on the directed graph based on the binomial tree structure.

In the upward pass, after each childless interval (corresponds to a node in the graph) collects data directly from different function values at the Chebyshev nodes, each coarser level interval receives and combines information from its two children intervals, and then sends the compressed information to its parent node. This process can be easily parallelized using *cilk_spawn* and *cilk_sync*, as shown by the pseudo-algorithm in Algorithm 1.

Algorithm 1 Upward Pass

```
1: function FIND_TRANSLATION_MATRIX(node C)
2:     if node C is childless then
3:         compute translation matrix directly from function values at Chebyshev nodes
4:     else
5:         find children nodes D and E of node C.
6:         cilk_spawn FIND_TRANSLATION_MATRIX(node D)
7:         cilk_spawn FIND_TRANSLATION_MATRIX(node E)
8:         cilk_sync
9:         construct C's translation matrix using children D and E's matrices
10:    end if
11: end function
```

In the downward pass, starting from the root interval with given constants $\lambda_l^B = 0$ and $\lambda_r^B = 0$, each child interval computes its coupling coefficients using its parent's coupling coefficients and the computed transformation matrix from the upward pass. This recursive procedure is detailed by the pseudo-code in Algorithm 2.

Algorithm 2 Downward Pass

```
1: function FIND_COUPLING_COEFFICIENTS(node C)
2:     if node C is root level then
3:         λ_l^[a,c] = 0 and λ_r^[a,c] = 0
4:     else
5:         compute λ_l and λ_r using (a) translation matrix and (b) parent's coefficients
6:         find children nodes D and E of node C.
7:         cilk_spawn FIND_COUPLING_COEFFICIENTS(node D)
8:         cilk_spawn FIND_COUPLING_COEFFICIENTS(node E)
9:     end if
10: end function
```

We want to mention that the work-stealing scheduling scheme in Cilk provides a guaranteed parallelization performance as in the following analytical result from [102]: Assuming T_P is the

execution time using P processors, T_1 is the time when using one processor (referred to as the "work"), and T_∞ is the execution time when there are infinitely many processors (corresponds to the critical-path length, or computational depth), then the Cilk's work-stealing scheduler runs the computation in expected time $O(T_1/P + T_\infty)$.

3.2.7 Numerical examples

Turning point We consider the turning point problem

$$\epsilon u''(x) - xu(x) = 0,$$

with boundary conditions

$$u(-1) = 1; \quad u(1) = 1,$$

has smooth regions, boundary layers, internal layers, and regions with dense oscillations. The exact solution is a linear combination of Airy functions

$$u(x) = c_1 Ai\left(\frac{x}{\sqrt{\epsilon}^3}\right) + c_2 Bi\left(\frac{x}{\sqrt{\epsilon}^3}\right).$$

The $\epsilon = 10^{-6}$ is chosen. The parallel performance is summarized in the Table 3.2.7. We observe a reasonable parallel efficiency is obtained.

Number of core	Computation times	parallel efficiency
1	2.57	100 %
2	1.38	93.1 %
4	0.74	86.8 %
8	0.43	74.7 %

3.3 Heterogeneous fast multipole method

In the acoustic and electromagnetism applications, Helmholtz equation is usually considered. Specifically, we consider the 2-D Helmholtz equation in free-space

$$(\Delta + \omega^2)u(\vec{x}) = 0$$

with the Sommerfeld radiation condition at ∞

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial}{\partial r}u(\vec{x}) - i\omega u(\vec{x})\right) = 0,$$

where $\vec{x} = (x, y)$ and $r = \|\vec{x}\|$.

In layered media, an impedance boundary condition is imposed on the interface defined by $y = 0$,

$$\frac{\partial u}{\partial \vec{n}} - i\alpha u = 0.$$

In the application of layered media problem, it is desirable to obtain a fast summation method to calculate the impedance Green's function

$$\phi(\vec{x}) = \sum_{j=1}^{n} q_j G(\omega\|\vec{x} - \vec{x}_j\|).$$

In the following sections, we consider the analytical representations of the Green's function to the Helmholtz equation with impedance boundary condition and present a hierarchical method to efficiently calculate its summation. The detailed derivation of the representation of the Green's function can be found in [58].

3.3.1 Spectral representation of the Green's function

The solution g_ω to the Helmholtz equation

$$-(\Delta + \omega^2)g_\omega(\vec{x}, \vec{x}_0) = \delta(\vec{x} - \vec{x}_0),$$

in an infinite homogeneous medium is referred to as the free-space Green's function, where $\vec{x} = (x, y) \in \mathbb{R}^2$, and $\delta(\vec{x} - \vec{x}_0)$ represents the Dirac delta function centered at $\vec{x}_0$. It is well known that

$$g_\omega(\vec{x}, \vec{x}_0) = \frac{i}{4}H_0^{(1)}(\omega\|\vec{x} - \vec{x}_0\|)$$

where $H_0^{(1)}$ denotes the zeroth-order Hankel function of the first kind. These Green's function satisfy the outgoing Sommerfeld radiation condition

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial}{\partial r} g_\omega(\vec{x},\vec{x}_0) - i\omega g_\omega(\vec{x},\vec{x}_0)\right) = 0,$$

where $r = \|\vec{x} - \vec{x}_0\|$. By Fourier calculation, the Green's function can be written as

$$g_\omega(\vec{x},\vec{x}_0) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{e^{i(\lambda_x(x-x_0)+\lambda_y(y-y_0))}}{\lambda_x^2+\lambda_y^2-\omega^2}\, d\lambda_x\, d\lambda_y.$$

Evaluating the integral in λ_y via contour deformation yields the expansion in plane waves (Sommerfeld integral):

$$g_\omega(\vec{x},\vec{x}_0) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{e^{-\sqrt{\lambda^2-\omega^2}|y-y_0|}}{\sqrt{\lambda^2-\omega^2}} e^{i\lambda(x-x_0)}\, d\lambda. \tag{3.3.1}$$

This representation is often referred to as the Sommerfeld identity, which can be separated into the propagating and evanescent modes for wave number variable $|\lambda| < k$ (propagating modes) and $|\lambda| > k$ (evanescent modes as $|y| \to \infty$), respectively, to arrive at the following form after some changes of variables

$$\begin{aligned} g_\omega(\vec{x},\vec{x}_0) &= g_\omega(\vec{x},\vec{x}_0)_{prop} + g_\omega(\vec{x},\vec{x}_0)_{evan} \\ &= \frac{i}{4\pi}\int_0^{\pi} e^{i\omega(|y-y_0|\sin(\theta)-(x-x_0)\cos(\theta))}\, d\theta \\ &+ \frac{1}{4\pi}\int_0^{\infty} \frac{e^{-t|y-y_0|}}{\sqrt{t^2+\omega^2}}\left(e^{i\sqrt{t^2+\omega^2}(x-x_0)} + e^{-i\sqrt{t^2+\omega^2}(x-x_0)}\right)\, dt. \end{aligned}$$

3.3.2 Complex image representation

We represent the image representation of the domain Green's function. The ansata is that the scattered field $u_\omega^s(\vec{x},\vec{x}_0)$ can be explicitly represented in the two-layered media as complex image contributions of the free-space kernel as

$$u_\omega^s(\vec{x},\vec{x}_0) = \int_0^{\infty} g_\omega(\vec{x},\vec{x}_0^{im} - s\widehat{y})\tau(s)\, ds,$$

where $\vec{x}_0^{im} = (x_0, -y_0)$, $\widehat{y} = (0, 1)$, and $\tau(s)$ is the complex image charge density distribution. By applying the impedance boundary condition, the image function $\tau(s)$ can be analytically solved by

$$\tau(s) = \delta(s) + \mu(s), \ s > 0,$$

where a point image is indicated by the Dirac delta distribution $\delta(s)$ and a line image $\mu(s)$ is complex and

$$\mu(s) = 2i\alpha e^{i\alpha s}.$$

As a result, we have

$$u_\omega^s(\vec{x}, \vec{x}_0) = g_\omega(\vec{x}, \vec{x}_0^{im}) + \int_0^\infty g_\omega(\vec{x}, \vec{x}_0^{im} - s\widehat{y})\mu(s)\ ds, \tag{3.3.2}$$

where the first term on the right-hand side represents the contribution from the point-image source, and the second term represents the contribution from the line-images. Then the domain's Green's function can therefore be written as

$$\begin{aligned} u_\omega(\vec{x}, \vec{x}_0) &= g_\omega(\vec{x}, \vec{x}_0) + u_\omega^s(\vec{x}, \vec{x}_0) \\ &= g_\omega(\vec{x}, \vec{x}_0) + \left(g_\omega(\vec{x}, \vec{x}_0^{im}) + \int_0^\infty g_\omega(\vec{x}, \vec{x}_0^{im} - s\widehat{y})\mu(s)\ ds \right). \end{aligned}$$

3.3.3 Sommerfeld integral representation

For the integral representation by the method of images, the numerical quadratures have difficulties to efficiently solve the integral due to the oscillatory line image density $\mu(s) = 2i\alpha e^{i\alpha s}$.

To simplify the expression, the Sommerfeld identity (3.3.1) can be used.

$$\begin{aligned}
&\int_0^\infty g_\omega(\vec{x}, \vec{x}_0^{im} - s\widehat{y}) e^{i\alpha s}\, ds \\
&= \int_0^\infty \left[\frac{1}{4\pi} \int_{-\infty}^\infty \frac{e^{-\sqrt{\lambda^2-\omega^2}|y+y_0+s|}}{\sqrt{\lambda^2-\omega^2}} e^{i\lambda(x-x_0)}\, d\lambda \right] e^{i\alpha s}\, ds \\
&= \frac{1}{4\pi} \int_{-\infty}^\infty \frac{e^{-\sqrt{\lambda^2-\omega^2}(y+y_0)} e^{i\lambda(x-x_0)}}{\sqrt{\lambda^2-\omega^2}} \left[\int_0^\infty e^{-\sqrt{\lambda^2-\omega^2}s} e^{i\alpha s}\, ds \right] d\lambda \\
&= \frac{1}{4\pi} \int_{-\infty}^\infty \frac{e^{-\sqrt{\lambda^2-\omega^2}(y+y_0)+i\lambda(x-x_0)}}{\sqrt{\lambda^2-\omega^2}} \frac{1}{\sqrt{\lambda^2-\omega^2}-i\alpha}\, d\lambda.
\end{aligned}$$

After some simple algebra, for $y > 0$, the spectral domain representation for the scattered field is derived

$$u_\omega^s(\vec{x}, \vec{x}_0) = \frac{1}{4\pi} \int_{-\infty}^\infty \frac{e^{-\sqrt{\lambda^2-\omega^2}(y+y_0)}}{\sqrt{\lambda^2-\omega^2}} e^{i\lambda(x-x_0)} \frac{\sqrt{\lambda^2-\omega^2}+i\alpha}{\sqrt{\lambda^2-\omega^2}-i\alpha}\, d\lambda. \tag{3.3.3}$$

To write in more compact form, we define

$$\widehat{\sigma}(\lambda) = \frac{\sqrt{\lambda^2-\omega^2}+i\alpha}{\sqrt{\lambda^2-\omega^2}-i\alpha},$$

we then have

$$u_\omega^s(\vec{x}, \vec{x}_0) = \frac{1}{4\pi} \int_{-\infty}^\infty \frac{e^{-\sqrt{\lambda^2-\omega^2}y}}{\sqrt{\lambda^2-\omega^2}} e^{i\lambda x} e^{-\sqrt{\lambda^2-\omega^2}y_0} e^{-i\lambda x_0} \widehat{\sigma}(\lambda)\, d\lambda. \tag{3.3.4}$$

Note in there, $\widehat{\sigma}(\lambda)$ is independent of $\vec{x}$ and $\vec{x}_0$.

The heterogeneous fast multipole method shares lot of similarity with the original fast multipole method. In particular, they have the identitcal multipole to multipole and local to local formulas, and the formation of multipole method is only modified slightly. The discussion of the fast multipole method for free space Helmholtz equation in 2-D can be found in [108].

3.3.4 Compression of data

Due to the smoothness of the far-field expansion of the Green's function, the observation is that when target boxes and source boxes are well-separated as in Figure 3.2, then the interaction between them is low-rank. In the 2-D FMM for the Helmholtz equation, far field expansions are based on

Graf's addition theorem, via the following formula, which is a trivial consequence of it; the formula expresses as a series the field at one point due to a unit source at another. Letting $\vec{x} = (\rho, \theta)$ and $\vec{x}_j = (\rho_j, \theta_j)$ in polar coordinates, with $\rho > \rho_j$, we have

$$H_0^{(1)}(\omega\|\vec{x} - \vec{x}_j\|) = \sum_{k=-\infty}^{\infty} H_k(\omega\rho)e^{ik\theta}(e^{-ik\theta_j}J_k(\omega\rho_j)).$$

The following theorem is an immediate consequence.

Theorem 3.3.1. *Suppose that the function $\phi : \mathbb{R}^2 \to \mathbb{C}$ is given by the formula*

$$\phi(\vec{x}) = \sum_{j=1}^{n} s_j H_0(\omega\|\vec{x} - \vec{x}_j\|),$$

so that $\|\vec{x} - \vec{x}_c\| \geq \|\vec{x}_j - \vec{x}_c\|$ for all $j = 1, \ldots, n$, where $\vec{x}_c$ is the center of $\vec{x}_j$. Then, denoting the polar coordinates of each point $\vec{x}_j - \vec{x}_c$ by (ρ_j, θ_j), and denoting the polar coordinates of the point $\vec{x} - \vec{x}_c$ by (ρ, θ),

$$\phi(\rho, \theta) = \sum_{k=-\infty}^{\infty} a_k H_k(\omega\rho)e^{ik\theta}, \tag{3.3.5}$$

where the coefficients $\{a_k\}$ are given by the formula

$$a_k = \sum_{j=1}^{n} s_j e^{-ik\theta_j} J_k(\omega\rho_j). \tag{3.3.6}$$

We observe that when the original source $\vec{x}_j$ in Figure 3.3 are well-separated from the target point $\vec{x}$, all the corresponding point-images $\vec{x}_j^{im}$ and the set of line-images on the rays are also

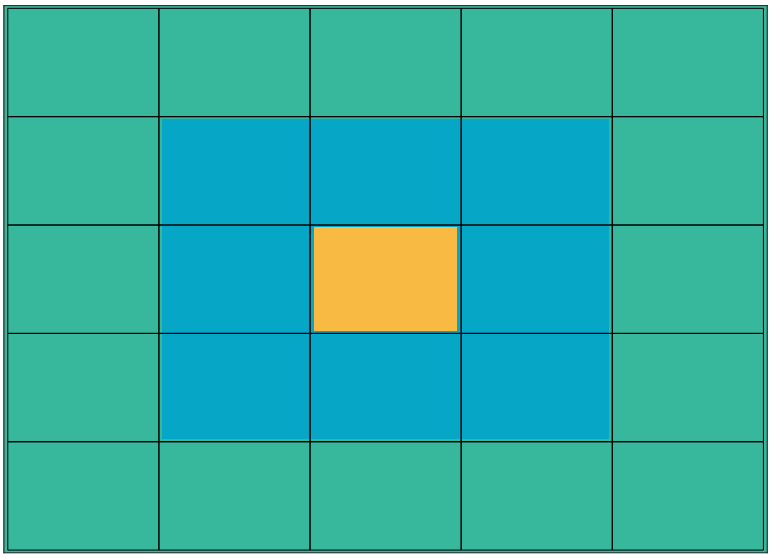


Figure 3.2: Yellow box is the target, green boxes are the well-separated boxes

well-separated from $\vec{x}$. Then by simple calculation, we calculate that

$$\begin{aligned} u^s_\omega(\vec{x}) &= \sum_{j=1}^{n} q_j u^s_\omega(\vec{x}, \vec{x}_j) \\ &= \frac{i}{4} \sum_{j=1}^{N} q_j \left(H_0^{(1)}(\omega \|\vec{x} - \vec{x}_j^{im}\|) + \int_0^\infty H_0^{(1)}(\omega \|\vec{x} - (\vec{x}_j^{im} - s\widehat{y})\|) \mu(s)\ ds \right) \\ &= \frac{i}{4} \sum_{p=-\infty}^{\infty} \overline{\alpha}_p \left(H_p^{(1)}(\omega \|\vec{x} - \vec{x}_c^{im}\|) e^{ip\theta_{im}} + \int_0^\infty H_p^{(1)}(\omega \|\vec{x} - (\vec{x}_c^{im} - s\widehat{y})\|) e^{ip\widehat{\theta}_{im}} \mu(s)\ ds \right), \end{aligned} \tag{3.3.7}$$

where $\vec{x}_j = (x_j, -y_j)$ are the coordinates of the point-image charge, $\overline{\alpha}_p$ is the complex conjugate the of free space multipole coefficient α_p in Eq. (3.3.5), and

θ_{im} is the polar angle of complex number $\vec{x} - \vec{x}_c^{im}$,

$\widehat{\theta}_{im}$ is the polar angle of complex number $\vec{x} - (\vec{x}_c^{im} - s\widehat{y})$.

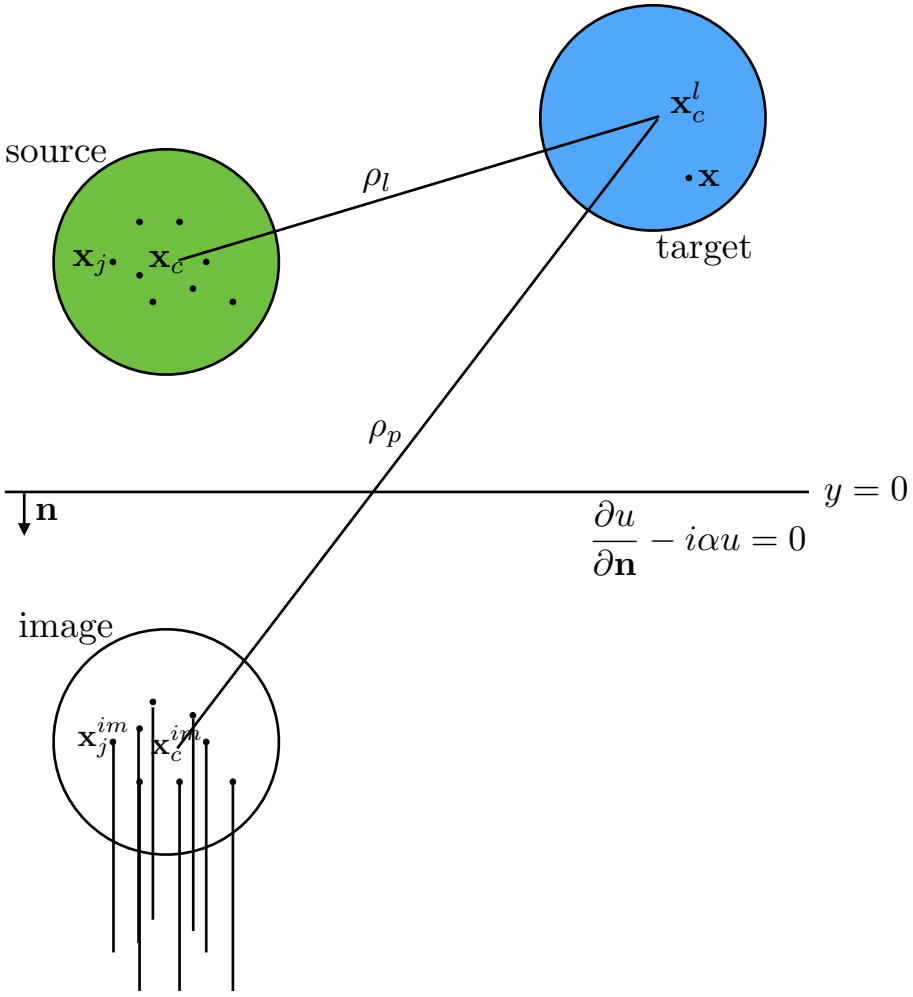


Figure 3.3: Impedance half-space and notation

To add the contribution from the original source, we obtain

$$u_\omega(\vec{x}) = \frac{i}{4} \sum_{p=-\infty}^{\infty} \alpha_p H_p^{(1)}(\omega \|\vec{x} - \vec{x}_c\|) e^{ip\theta_c} \tag{3.3.8}$$

$$+ \frac{i}{4} \sum_{p=-\infty}^{\infty} \overline{\alpha}_p \left(H_p^{(1)}(\omega \|\vec{x} - \vec{x}_c^{im}\|) e^{ip\theta_{im}} + \int_0^\infty H_p^{(1)}(\omega \|\vec{x} - (\vec{x}_c^{im} - s\widehat{y})\|) e^{ip\widehat{\theta}_{im}} \mu(s)\, ds \right). \tag{3.3.9}$$

3.3.5 Translation of data

We then derive the translation rules for the data. Since the multipole coefficients are very similar to the free space case, the multipole to multipole and local to local formulas can follow the classical

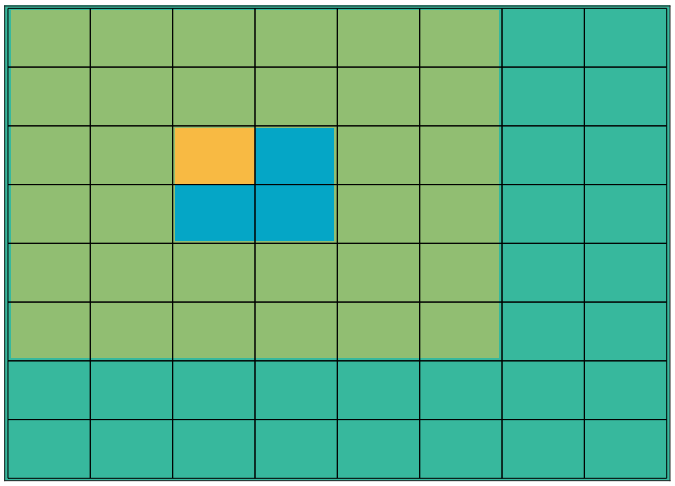


Figure 3.4: Yellow box is the target box; blue boxes along with yellow box is its parent; green boxes are well-separated from them.

FMM.

Children to parent The observation is that when source boxes are well-separated from the target box and its parent as in Figure 3.4, then then information can be simply translated from the target box to its parent box.

Theorem 3.3.2 (Multipole to Multipole)**.** *Suppose that*

$$\phi(\vec{x}) = \sum_{k=-\infty}^{\infty} a_k H_k(\omega\rho_{c0}) e^{ik\theta_{c0}}$$

is a multipole expansion centered at $\vec{x}_{c0}$ *and* $\vec{x}_j - \vec{x}_{c0} = (\rho_{c0}, \theta_{c0})$*. Then for a new well separated box from* $\vec{x}$ *centered at* $\vec{x}_{c1}$ *with* $\vec{x}_j - \vec{x}_{c1} = (\rho_{c1}, \theta_{c1})$*,*

$$\phi(\vec{x}) = \sum_{m=-\infty}^{\infty} b_m H_m(\omega\rho_{c1}) e^{ik\theta_{c1}},$$

where

$$b_m = \sum_{j=-\infty}^{\infty} e^{-ij(\theta_{01}-\pi)} a_{m-j} J_j(\omega\rho_{01}),$$

with $\rho_{01} = \|\vec{x}_{c1} - \vec{x}_{c0}\|$ *and* θ_{01} *is the angle between* $\vec{x}_{c1} - \vec{x}_{c0}$.

Parent to child The derivation of the local expansion to local expansion is similar to the translation of multipole coefficients and the results are also taken from the classical FMM.

Theorem 3.3.3 (Local to Local)**.** *Suppose that*

$$\phi(\vec{x}) = \sum_{k=-\infty}^{\infty} a_k J_k(\omega\rho_{c0}) e^{ik\theta_{c0}},$$

with $\vec{x} - \vec{x}_{c0} = (\rho_{c0}, \theta_{c0})$ *and* $\vec{x} - \vec{x}_{c1} = (\rho_{c1}, \theta_{c1})$*, where* $\vec{x}_{c0}$ *is the original center of the local expansion and* $\vec{x}_{c1}$ *is the new center of the local expansion, then the translated local expansion can be represented as*

$$\phi(\vec{x}) = \sum_{m=-\infty}^{\infty} b_m J_m(\omega\rho_{c1}) e^{ik\theta_{c1}},$$

where

$$b_m = \sum_{j=-\infty}^{\infty} e^{-ij\theta_{01}} a_{m-j} J_j(\omega\rho_{01}),$$

where $\rho_{01} = \|\vec{x}_{c1} - \vec{x}_{c0}\|$ *and* θ_{01} *is the angle between* $\vec{x}_{c1} - \vec{x}_{c0}$.

3.3.6 Conversion of data

We now have translation tools for information spread between children and parents. Yet the base for the expression of the data is different in the translation, and from parent to children, there are certain information which cannot be translated directly, hence the multipole to local translation is derived. The translation here is focused on the interation list as in Figure 3.5.

For the “multipole" expansion

$$u^s_\omega(\vec{x}) = \frac{i}{4} \sum_{p=-\infty}^{\infty} \overline{\alpha}_p \left(H_p(\omega\|\vec{x} - \vec{x}_c^{im}\|) e^{ip\theta_{im}} + \int_0^\infty H_p(\omega\|\vec{x} - (\vec{x}_c^{im} - s\widehat{y})\|) e^{ip\widehat{\theta}_{im}} \mu(s)\, ds \right),$$

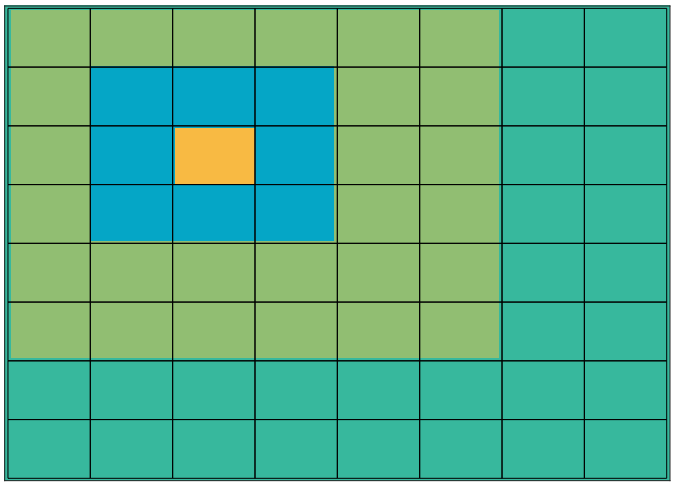


Figure 3.5: Yellow box is the source box and the light green is its interaction list.

by Graf's addition theorem, we calculate

$$\begin{aligned} u^s_\omega(\vec{x}) &= \frac{i}{4}\sum_{p=-\infty}^{\infty}\sum_{m=-p}^{p}\overline{\alpha}_m\Big(H_{m-p}(\omega\widetilde{\rho}_{im})J_p(\omega\|\vec{x}-\vec{x}^l_c\|)e^{i(m-p)\widetilde{\theta}_{im}}e^{ip\theta} \\ &\quad + \int_0^\infty H_{m-p}(\omega\widehat{\widetilde{\rho}}_{im})J_p(\omega\|\vec{x}-\vec{x}^l_c\|)e^{i(m-p)\widehat{\widetilde{\theta}}_{im}}\mu(s)e^{ip\theta}\ ds\Big) \\ &= \frac{i}{4}\sum_{p=-\infty}^{\infty}\beta^s_p J_p(\omega\|\vec{x}-\vec{x}^l_c\|)e^{ip\theta}, \end{aligned} \tag{3.3.10}$$

where the local expansion coefficients are given by

$$\beta^s_p = \sum_{m=-\infty}^{\infty}\overline{\alpha}_m\left(H_{m-p}(\omega\widetilde{\rho}_{im})e^{i(m-p)\widetilde{\theta}_{im}} + \int_0^\infty H_{m-p}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-p)\widehat{\widetilde{\theta}}_{im}}\mu(s)\ ds\right), \tag{3.3.11}$$

with θ the polar angle of $\vec{x}-\vec{x}^l_c$, and

$$\begin{aligned} &(\widetilde{\rho}_{im},\widetilde{\theta}_{im}) \text{ are the polar coordinates of } \vec{x}^l_c - \vec{x}^{im}_c, \\ &(\widehat{\widetilde{\rho}}_{im},\widehat{\widetilde{\theta}}_{im}) \text{ are the polar coordinates of } \vec{x}^l_c - (\vec{x}^{im}_c - s\widehat{y}). \end{aligned}$$

The local expansion for the total field $u(\vec{x})$ is simply the sum of the free space Green's function and scattered field local expansions. As the translation operator from the compressed "multipole

coefficients" $\{\alpha_p\}$ to the local coefficients $\{\beta_p^s\}$ involves the complex conjugate operator, for notation reasons, instead of combining the free-space with the complex image contributions in one single translation, we only construct the mapping matrix A for the scattered field,

$$\beta_p^s = \sum_{m=-p}^{p} A_{p,m}\overline{\alpha}_m,$$

where

$$A_{p,m} = H_{m-p}(\omega\widetilde{\rho}_{im})e^{i(m-p)\widetilde{\theta}_{im}} + \int_0^\infty H_{m-p}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-p)\widehat{\widetilde{\theta}}_{im}}\mu(s)\ ds.$$

However, this integrand is highly oscillatory for large s. We want to simplify the integration. The main relationship we want to use is

$$H_n(\omega\rho)e^{in\theta} = \left(-\frac{1}{\omega}\right)^n \left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)^n H_0(\omega\rho).$$

Applying the identity to the spectral representation of the Hankel function (3.3.1), we have

$$H_n(\omega\rho)e^{in\theta} = \frac{(-i)^n}{i\pi}\int_{-\infty}^{\infty} \frac{e^{-\sqrt{\lambda^2-\omega^2}y}}{\sqrt{\lambda^2-\omega^2}} e^{i\lambda x}\left(\frac{\lambda - \sqrt{\lambda^2-\omega^2}}{\omega}\right)^n\ d\lambda.$$

Using the change of variable technique to separate it to the propagating part and evanescent part, we reformulate it to be

$$\begin{aligned} H_{m-p}(\omega\widetilde{\rho}_{im})e^{i(m-p)\widetilde{\theta}_{i}m} = {} & \frac{i^{m-p}}{\pi}\int_0^\pi e^{i\omega(y\sin(\tau)-x\cos(\tau))}e^{-i(m-p)\theta}\ d\tau \\ & + \frac{(-i)^{m-p}}{i\pi}\int_0^\infty \frac{e^{-ty}}{\sqrt{t^2+\omega^2}}K(t)\ dt, \end{aligned}$$

where

$$K(t) = e^{i\sqrt{t^2+\omega^2}x}\left(\frac{\sqrt{t^2+\omega^2}-t}{\omega}\right)^{m-p} + e^{-i\sqrt{t^2+\omega^2}x}\left(\frac{-\sqrt{t^2+\omega^2}-t}{\omega}\right)^{m-p}.$$

With some simple algebraic manipulation, we deduce that

$$\begin{aligned} A_{p,m} =& \frac{i^{m-p}}{\pi} \int_0^{\pi} e^{i\omega(y\sin\tau - x\cos\tau)} e^{-i(m-p)\theta} \left(\frac{\omega\sin(\tau)-\alpha}{\omega\sin(\tau)+\alpha} \right) d\tau \\ &+ \frac{(-i)^{m-p}}{i\pi} \int_0^{\infty} \frac{e^{-ty}}{\sqrt{t^2+\omega^2}} \left(e^{i\sqrt{t^2+\omega^2}x} \left(\frac{\sqrt{t^2+\omega^2}-t}{k} \right)^{m-p} \right. \\ &\left. + e^{-i\sqrt{t^2+\omega^2}x} \left(\frac{-\sqrt{t^2+\omega^2}-t}{k} \right)^{m-p} \right) \left(\frac{t+i\alpha}{t-i\alpha} \right) dt. \end{aligned}$$

Now for the propagating part, since the interval is finite, high-order Gaussian quadrature is applied. For evanescent term, the generalized Laguerre quadrature with weight function $t^n e^{-t}$ is used. Both numerical schemes work very efficiently so that we consider the computational complexity for this numerical integration is constant time.

In the numerical implementation, the matrix can be either computed on-the-fly using high-order Gauss and Laguerre quadratures or precomputed and stored. We can estimate the required storage in the algorithm for the precomputation. For the box with a fixed x and y, i.e., the information that is due to a well-separated box in the interaction list, the number of entries we need to store in $A_{p,m}(x,y)$ is only $4p$, since it is not a two variable function of m and p and only depends on $m-p$. For each box, there are at most 27 boxes in the interaction list. For the two-layered media case, the matrix also depends on the y-coordinate of the center of the box as the translation operator takes different values as their images change. Thus, we can conclude that at tree level l, there are 2^l different values of y-coordinate, and for each y-coordinate 27 possible well-separated boxes that require $4p$ complex values. Therefore, the total required storage for a system with L-levels is approximately $(2^{L+1} \cdot 27 \cdot 4p) \cdot 16$ bytes. For comparison, there are approximate 4^{L+1} boxes with $2p$ entries in the multipole expansion we need to store.

3.3.7 Accelerating evaluation of local direct interactions

In this section, we consider the direct evaluation of the local particles. For a source box with N particles located at $\{\vec{x}_j\}$, its domain Green's function contribution to a target point $\vec{x}$ in a

neighboring box is defined as

$$u_\omega^d(\vec{x}) = \frac{i}{4}\sum_{j=1}^{N} q_j H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j\|) + \frac{i}{4}\sum_{j=1}^{N} q_j \left(H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j^{im}\|) + \int_0^\infty H_0^{(1)}(\omega\|\vec{x}-(\vec{x}_j^{im}-s\widehat{y})\|)\mu(s)\ ds \right).$$

In this formula, it is possible to take advantage of the compressed scattered field representations of the domain Green's function so that it can be evaluated more efficiently.

In Figure 3.6, we show a source box sitting next to a target box. In this figure, we observe that, most part of the line images are well-separated from the target box. Therefore, we choose an approapriate constant C so that the line-images is divided into two parts: those that are well-separated from the target box and those that are not. Specifically, we write $u_\omega^d(\vec{x}) = I + II$ where

$$I = \frac{i}{4}\sum_{j=1}^{N} q_j H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j\|) \tag{3.3.12}$$

$$+ q_j \left(H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j^{im}\|) + \int_0^C H_0^{(1)}(\omega\|\vec{x}-(\vec{x}_j^{im}-s\widehat{y}))\mu(s)\ ds \right), \tag{3.3.13}$$

$$II = \frac{i}{4}\sum_{j=1}^{N} q_j \int_C^\infty H_0^{(1)}(\omega\|\vec{x}-(\vec{x}_j^{im}-s\widehat{y})\|)\mu(s)\ ds. \tag{3.3.14}$$

The integral in I is computed directly by using high order quadrature for its finite size. On the other hand, since $\vec{x}_j^{im} - s\widehat{y}$ is well-separated from the target box, we could again compress the Hankel

function so that

$$
\begin{aligned}
II &= \frac{i}{4}\sum_{j=1}^{N} q_j \int_C^{\infty} H_0^{(1)}(\omega\|\vec{x}-(\vec{x}_j^{im}-s\widehat{y})\|)\mu(s)\ ds \\
&= \frac{i}{4}\int_C^{\infty} \sum_{m=-\infty}^{\infty} \overline{\alpha}_m H_m^{(1)}(\omega\|\vec{x}-(\vec{x}_c^{im}-s\widehat{y})\|)e^{im\widehat{\theta}_{im}}\mu(s)\ ds \\
&= \frac{i}{4}\int_C^{\infty} \sum_{m=-\infty}^{\infty} \overline{\alpha}_m \sum_{n=-\infty}^{\infty} H_{m-n}^{(1)}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-n)\widehat{\widetilde{\theta}}_{im}} J_n(\omega\|\vec{x}-\vec{x}_c^{l}\|)e^{in\theta}\mu(s)\ ds \\
&= \frac{i}{4}\sum_{n=-\infty}^{\infty}\left(\sum_{m=-\infty}^{\infty} \overline{\alpha}_m \int_C^{\infty} H_{m-n}^{(1)}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-n)\widehat{\widetilde{\theta}}_{im}}\mu(s)\ ds\right) J_n(\omega\|\vec{x}-\vec{x}_c^{l}\|)e^{in\theta} \\
&= \frac{i}{4}\sum_{m=-\infty}^{\infty} L_n J_n(\omega\|\vec{x}-\vec{x}_c^{l}\|)e^{in\theta},
\end{aligned}
$$

where

$$
L_n = \sum_{m=-\infty}^{\infty} \overline{\alpha}_m \int_C^{\infty} H_{m-n}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-n)\widehat{\widetilde{\theta}}_{im}}\mu(s)\ ds = \sum_{m=-\infty}^{\infty} \overline{\alpha}_m B_{m,n}
$$

and the translation matrix is given by

$$
B_{m,n} = \int_C^{\infty} H_{m-n}^{(1)}(\omega\widehat{\widetilde{\rho}}_{im})e^{i(m-n)\widehat{\widetilde{\theta}}_{im}}\mu(s)\ ds,
$$

which can be evaluated efficiently by using the corresponding Sommerfeld integral representation.

In the tree structure, most boxes are well-separated from the interface $y = 0$, implying that $C = 0$ for most direct interactions of the source and target boxes,

$$
\begin{aligned}
I &= \frac{i}{4}\sum_{j=1}^{N} q_j H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j\|), \\
II &= \frac{i}{4}\sum_{j=1}^{N} q_j \left(H_0^{(1)}(\omega\|\vec{x}-\vec{x}_j^{im}\|) + \int_0^{\infty} H_0^{(1)}(\omega\|\vec{x}-(\vec{x}_j^{im}-s\widehat{y})\|)\mu(s)\ ds\right).
\end{aligned}
$$

3.3.8 Algorithm

Now we present the pseudo-code for our algorithm.

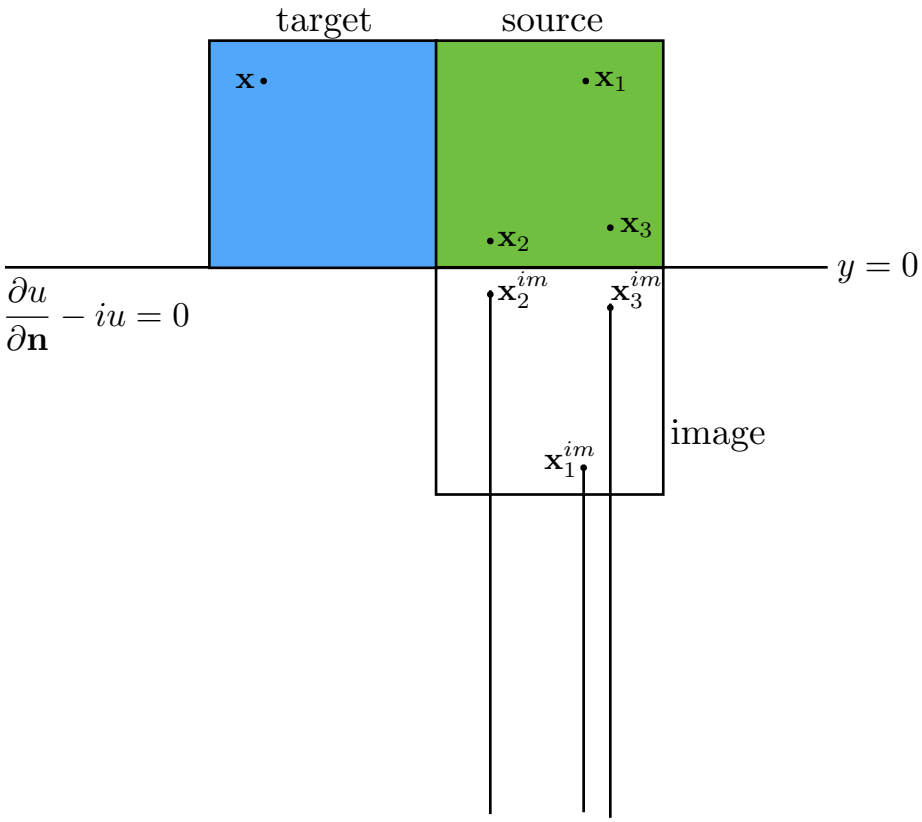


Figure 3.6: Images are separated to near- and far-field by choosing appropriate C.

Heterogeneous 2-D FMM for Two-layered Media with Impedance Boundary Conditions

Step 1: Initialization

Generate an adaptive hierarchical tree structure and precompute tables.

Comment [L denotes the maximum refinement level in the adaptive tree determined by a prescribed number s representing the maximum allowed number of particles in a childless box. **]**

Step 2: Upward Pass

for $l = L, \ldots, 0$

  for all boxes j on level l

    if j is a leaf node

      form the *free-space* multipole expansion using Eq. (3.3.5).

    else

      form the *free-space* multipole expansion by merging children's

        expansions using the *free-space* "multipole-to-multipole"

        translation operator.

    endif

  end

end

Cost [All operations in this step are the same as those in the *free-space* FMM. **]**

Step 3: Downward Pass

for $l = 1, \ldots, L$

  for all boxes j on level l

    shift the local expansion of j's parent to j itself using the *free-space*

      "local-to-local" translation operator.

    collect interaction list contribution using the precomputed table and

the "multipole-to-local" translation operator in Eq. (3.3.10).

end

end

Cost [Using the precomputed table, the cost is expected to be the same as in the *free-space* FMM. Overhead operations are required when tables are computed on-the-fly. **]**

Step 4: Evaluate Local Expansions

for each leaf node (childless box)

collect part II in Eq. (3.3.14) from neighboring (including self) boxes.

evaluate the local expansion at each particle location.

end

Comment [At this point, for each target point, its far field contribution (including those from well-separated images) has been computed. **]**

Cost [Compared with the *free-space* FMM, additional translations are required to translate the multipole expansions of images to local expansions. The heterogeneous translation operators can be computed on-the-fly or precomputed. The amount of work is constant for each leaf node. **]**

Step 5: Local Direct Interactions

for $i = 1, ..., N$

compute Part I in Eq. (3.3.13) of target point i with original and image sources in the neighboring boxes.

end

Cost [When the computational domain is well-separated from the boundary $y = 0$, this step only involves the evaluation of the *free-space* kernel and the cost is the same as the *free-space* FMM. When the computational domain is close to the boundary $y = 0$, a constant number of additional operations are required for each i in a very small subset of the particles to evaluate the near-field point- and line-image contributions from Part I in (3.3.13)). **]**

Table 3.1: CPU time (seconds) for different N using $p = 39$ and $k = 0.1$

N	100	6400	10000	90000	360000	640000	810000	1000000
CPU time	0.01	0.67	1.19	10.92	46.58	100.85	116.03	135.05

3.3.9 Numerical example

We present some preliminary numerical results in this section to demonstrate the performance of the new heterogeneous FMM algorithm for the two-layered media with the interface placed at $y = 0$, and set $\alpha = 1$ in the impedance boundary condition. We assume the source and target points are the same set of N particles located in a unit box centered at $(0, 1.5)$ as shown in Figure 3.7. The numerical simulations are performed on a desktop with 3.7 GHz Xeon E5 processor and 32 GB RAM using the gcc compiler version 4.9.3. All the required translation tables are precomputed using Mathematica.

Since the analytical solution is not availabel, we check the error by refinement. We also demonstrate the algorithm efficiency by presenting the CPU times in the table 3.1 for different numbers of source/target points N from 100 to 1,000,000 for $\omega = 0.1$. One can see the clearly linear scaling of the heterogeneous FMM algorithm. We also compare it to the directly computation.

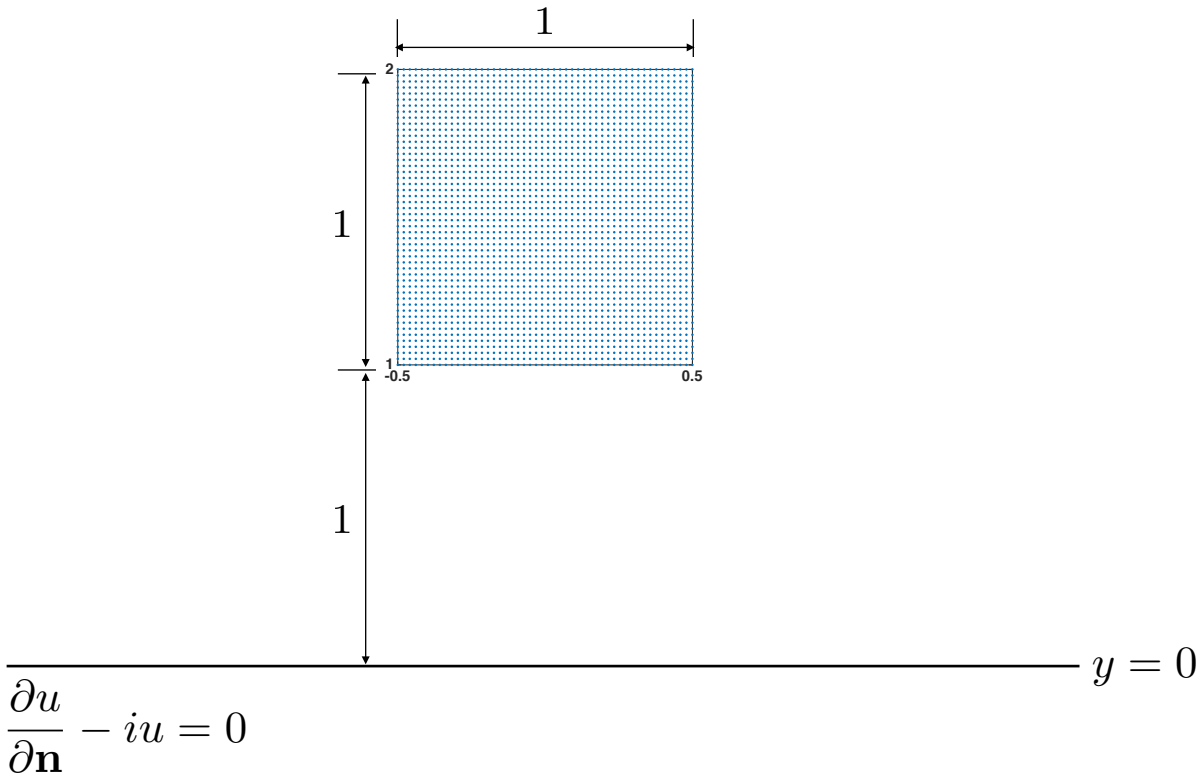


Figure 3.7: Uniform distribution in a unit square on top of half-space

Figure 3.8: CPU time (seconds) for different N using $p = 39$ and $\omega = 0.1$

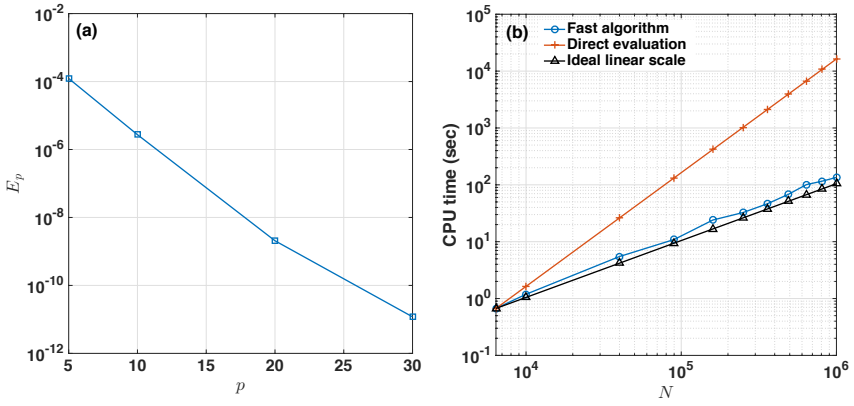


Figure 3.9: (a) Convergence and
(b) linear CPU time scaling for the impedance half-space problem with $\omega = 0.1$.

CHAPTER 4

Generalizations and Concluding Remarks[1]

In this dissertation, we have:

1. described the formulations of the PDEs for the electromagnetic interactions in the materials and several relaxations of the models under some conditions (Chapter 1);

2. improved the performance of deferred correction method by some new preconditioning techniques and proposed a well-conditioned integral equation method for the efficient solution in the temporal direction (Chapter 2); and

3. presented the framework of the hierarchical methods in the spatial direction, implemented the parallel version of the fast recursive solver for two-point boundary value problems, and developed the heterogeneous FMM to layered media Helmholtz equation (Chapter 3).

Our main contribution is the (pseudo-) spectral method, which is accurate and can treat a wide variety of time-dependent differential equations, and the hierarchical method including the heterogeneous FMM, which is extremely fast to the layered media problem. Thus, they should prove useful tools in many areas of computational science and engineering.

The principle feature of the SDC is its extremely accurate resolution of the solution and slow growth in the long-time simulation, the step-size often beating classical techniques. Hence, our algorithm is particularly efficient for chaotic systems and long-time dynamics; such is the case with the calculation of TDKS equation in TDDFT, and our results reflect its efficiency in this regard. Still, our scheme can become unwidely for complicated problems due to the lack of an efficient

[1]Some materials in this chapter previously appeared as an article in the arXiv and is already submitted. The original citation is as follows: M. Cho, J. Huang, D. Chen, and W. Cai. "A heterogeneous FMM for 2-D layered media Helmholtz equation I: two & three cases", arXiv preprint arXiv:1703.09136 (2017). Some materials in this chapter will appear in the future and are currently in preparation.

preconditioner. In many cases, the Euler's preconditioning is not sufficiently efficient. Thus, an outstanding issue is to a development of preconditioning techniques in different scenarios.

For the hierarchical methods, we have developed a heterogeneous FMM for the two-layered media problems. However, the current implementation heavily relies on the method of image representation, which is mostly unavailable with more than two layers. Thus, it is important to derive an efficient compression scheme for the Sommerfeld formulation, which is available for multi-layered media problems.

In this concluding chapter, we offer some insights toward those concerns. We begin by discussing the optimizing procedure to train an "optimal" preconditioner with more considerations. We then give some brief comments on how an integral equation method may be achieved for the heat kernel with Dirichlet boundary conditions. Such solvers for complicated boundary conditions are expected to have a tremendous impact, especially in biophysical systems. Finally, we discuss how the heterogeneous FMM can be generalized to three-layered media problems, as well as how similar ideas might be used to multi-layered media problems. We end with some final remarks and perspectives for the future.

4.1 More considerations on "optimal" preconditioner

Similar to the fast direct solver [109], the idea of the "optimal" preconditioner relies on that the computational complexity is very expensive, thus we are willing to do many jobs in advance to accelerate the calculation in the precomputation stage. Analog to the machine learning methods, there is much information we already know before the time propagation yet we don't utilize, including the complex geometry, the stiffness of the operators, boundary conditions and so on. By using all information in advance, it is possible to train an "optimal" preconditioner. Recall from Section 2.5 that we have only shown the efficiency of the "optimal" preconditioner in 1-D simple cases and focus on the low-order linear multistep methods. A nature step after is to incorporate other preconditioners, consider geometry and boundary conditions, train parameters in the iterative scheme and so on.

4.2 Integral equation method for general cases

We have discussed the integral equation method in the temporal direction, primary for the periodic boundary conditions. The idea can be generalized to other boundary conditions, and possibly complicate geometry coupled with potential theory. We want to illustrate an example how the generalization can be achieved. Consider the heat equation with Dirichlet zero boundary

condition with $x \in [-1, 1]$, a Fourier series calculation shows that

$$G(x,t;y,\tau) = \frac{1}{\pi}\sum_{n=1}^{\infty} e^{-\frac{\pi^2}{4}(t-\tau)} \sin\left(\frac{n}{2}\pi x\right)\sin\left(\frac{n}{2}\pi y\right).$$

On the other hand, the method of images can be used to show that

$$G(x,t;y,\tau) = \frac{1}{4\pi(t-s)}\sum_{n\in\mathbb{Z}}\sum_{\sigma=\pm 1} \sigma e^{-\frac{(x-\sigma y-2n)^2}{4(t-s)}}.$$

There have extensive study of rapid evaluation of Heat kernel in those forms [110, 111]. Both Green's function converge exponentially fast, but in different regions of time. Combining those two Green's functions, the calculation of the integral equation method can be carried out extremely fast.

4.3 Toward a heterogeneous FMM for multi-layered media Helmholtz equations

Although we have thus far been concerned only with the solution for two-layered media Helmholtz equation, similar ideas may prove useful for the multi-layered media case. For multi-layered media, the explicit forms of the complex image representations are in general unavailable and the domain's Green's functions are customarily expressed in terms of the Sommerfeld integrals [112, 113, 114, 115]. For simplicity, we consider a three-layered media problem with setting outlined in [59].

Assume a point source is located at $\vec{x}_0 = (x_0, y_0)$ in the top layer with a wave number ω_1 and the two interfaces of the three-layered media are located at $y = 0$ and $y = -d$, respectively. The Sommerfeld integral representation for the scattered field in the top layer ($y > 0$) can be represented as

$$u_1^s(\vec{x}) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{e^{-\sqrt{\lambda^2-\omega_1^2}y}}{\sqrt{\lambda^2-\omega_1^2}} e^{i\lambda x} e^{-\sqrt{\lambda^2-\omega_1^2}y_0} e^{-i\lambda x_0}\sigma_1(\lambda)\, d\lambda \tag{4.3.1}$$

where the unknown function $\sigma_1(\lambda)$ will be determined later. In the middle layer with a wave number ω_2, the scattered field u_2^s can be written as the sum of the contributions u_2^t (from upper interface)

and u_2^b (from lower interface) as

$$u_2^t(\vec{x}) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{e^{\sqrt{\lambda^2-\omega_2^2}y}}{\sqrt{\lambda^2-\omega_2^2}} e^{i\lambda x} e^{-\sqrt{\lambda^2-\omega_2^2}y_0} e^{-i\lambda x_0} \sigma_2^+(\lambda)\ d\lambda,$$
$$u_2^b(\vec{x}) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{e^{-\sqrt{\lambda^2-\omega_2^2}(y+d)}}{\sqrt{\lambda^2-\omega_2^2}} e^{i\lambda x} e^{-\sqrt{\lambda^2-\omega_2^2}y_0} e^{-i\lambda x_0} \sigma_2^-(\lambda)\ d\lambda,$$

and in the bottom layer with wave number ω_3, we have

$$u_3^s(\vec{x}) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{e^{\sqrt{\lambda^2-\omega_3^2}(y+d)}}{\sqrt{\lambda^2-\omega_3^2}} e^{i\lambda x} e^{-\sqrt{\lambda^2-\omega_3^2}y_0)} e^{-i\lambda x_0} \sigma_3(\lambda)\ d\lambda,$$

where $\sigma_2^+(\lambda), \sigma_2^-(\lambda)$, and $\sigma_3(\lambda)$ are unknown quantities which are associated layer reflection and transmission coefficients of waves in spectral domain. When the interface conditions are given by $[u] = 0$ and $\left[\frac{\partial u}{\partial n}\right] = 0$, these quantities can be determined by solving the linear system

$$\begin{bmatrix} \frac{1}{\sqrt{\lambda^2-\omega_1^2}} & \frac{-1}{\sqrt{\lambda^2-\omega_2^2}} & -\frac{e^{-\sqrt{\lambda^2-\omega_2^2}d}}{\sqrt{\lambda^2-\omega_2^2}} & 0 \\ 0 & \frac{e^{-\sqrt{\lambda^2-\omega_2^2}d}}{\sqrt{\lambda^2-\omega_2^2}} & \frac{1}{\sqrt{\lambda^2-\omega_2^2}} & \frac{-1}{\sqrt{\lambda^2-\omega_3^2}} \\ 1 & 1 & e^{-\sqrt{\lambda^2-\omega_2^2}d} & 0 \\ 0 & e^{-\sqrt{\lambda^2-\omega_2^2}d} & -1 & -1 \end{bmatrix} \begin{bmatrix} \sigma_1(\lambda) \\ \sigma_2^+(\lambda) \\ \sigma_2^-(\lambda) \\ \sigma_3(\lambda) \end{bmatrix} = \begin{bmatrix} \frac{-1}{\sqrt{\lambda^2-\omega_1^2}} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

When all the source and target points are located in the top layer, we see that the domain Green's function u_1^s is similar to Eq. (3.3.4) but with a different σ function, the nature of which will be revealed in the follows.

One additional complexity of the multi-layered media computation is that the source and target points may be located in different layers, which must be investigated further. Here, we discuss how we can generalized our idea applying to Eq. (4.3.1). We start from the Sommerfeld representation of $H_n(\omega\rho)e^{in\theta}$. Applying the relation

$$H_n(\omega\rho)e^{in\theta} = \left(-\frac{1}{k}\right)^n \left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)^n H_0(\omega\rho)$$

where (ρ, θ) are the polar coordinates of the complex under $x + iy$, and using the Sommerfeld

representation of $H_0(\omega\rho)$ given by Eq. (3.3.1), we have for $y > 0$,

$$H_n(\omega\rho)e^{in\theta} = \frac{(-i)^n}{i\pi}\int_{-\infty}^{\infty}\frac{e^{-\sqrt{\lambda^2-\omega^2}y}}{\sqrt{\lambda^2-\omega^2}}e^{i\lambda x}\left(\frac{\lambda-\sqrt{\lambda^2-\omega^2}}{\omega}\right)^n\ d\lambda.$$

Plugging this representation in Eq. (3.3.8), and integrating the s variable analytically, we have the compressed Sommerfeld representation directly as

$$u^s(\vec{x}) \approx \int_{-\infty}^{\infty}\frac{e^{-\sqrt{\lambda^2-\omega^2(y+y_c)}}}{\sqrt{\lambda^2-\omega^2}}e^{i\lambda(x-x_c)}\left(\frac{1}{4\pi}\sum_{p=-P}^{P}\overline{\alpha}_p(-i)^p\left(\frac{\lambda-\sqrt{\lambda^2-\omega^2}}{\omega}\right)^p\right)\left(\frac{\sqrt{\lambda^2-\omega^2}+i\alpha}{\sqrt{\lambda^2-\omega^2}-i\alpha}\right)\ d\lambda. \tag{4.3.2}$$

Comparing with uncompressed Sommerfeld representation by adding up the Sommerfeld representation of the scatter field in Eq. (3.3.3) for each source, we have

$$u^s(\vec{x}) = \int_{-\infty}^{\infty}\frac{e^{-\sqrt{\lambda^2-\omega^2}(y+y_c)}}{\sqrt{\lambda^2-\omega^2}}e^{i\lambda(x-x_c)}\left(\frac{1}{4\pi}\sum_{j=1}^{N}q_j e^{-\sqrt{\lambda^2-\omega^2}(y_j-y_c)}e^{i\lambda(x_c-x_j)}\right)\left(\frac{\sqrt{\lambda^2-\omega^2}+i\alpha}{\sqrt{\lambda^2-\omega^2}-i\alpha}\right)\ d\lambda. \tag{4.3.3}$$

We further notice that $\overline{\alpha}_p$ is independent of λ, and the compressed term in Eq. (4.3.2) is the Laurent expression in $z = \frac{\lambda-\sqrt{\lambda^2-\omega^2}}{\omega}$ of the uncompressed term in Eq. (4.3.3).

Comparing Eq. (4.3.2) with (4.3.3), we can identify the roles of different terms in the Sommerfeld representation of the domain Green's function. In particular, we see that the conjugate of the free-space multipole expansion coefficients are the same as the Laurent expansion ones in the compressed representation. This observation reveals how the domain Green's function can be compressed directly when the line image is unavailable. For the instance, for the top layer Sommerfeld domain Green's function in Eq. (4.3.1) for the three layered media setting, as the terms "free-space info" and "uncompressed" have the same structure as in Eq. (4.3.3), and the term "image info" is independent of $\vec{x}$ and $\vec{x}_0$, we can therefore simply compute the free-space multipole expansion either directly from the sources, or through the free-space "multipole-to-multipole" translations, and the results will directly give a compressed Sommerfeld representation similar to Eq. (4.3.2).

The error analysis of the direct compression of the Sommerfeld integral representation described in Eqs. (4.3.2) and (4.3.3) is not an easy task. Luckily for the 2-D half-space problem, the error

analysis becomes trivial when performed in the physical domain using the image representation. Note the error analysis only requires that the target box and all the image sources are well-separated, which can be easily carried out for the multi-layered case using repeated image reflections, without knowing the exact density and location of the image.

4.4 Conclusion

In this work, we have presented an accurate solver in the temporal direction and hierarchical methods in the spatial direction and have demonstrated their use in the numerical examples. For SDC method, the primary novelty is the improvement of the preconditioning techniques, especially in the stiff cases. For hierarchical methods, we have designed an extremely fast solver to calculate the layered media Helmholtz equation with the controlled error. We believe that our techniques will enable new large-scale physical simulations, both in space and time. Furthermore, because of the generality of the hierarchical approaches, our methods can also be applied to many other problems in computational science and data science. A number of algorithmic issues remain open, notably the efficient extension of the SDC to problems with complex geometries and boundary conditions, and the heterogeneous FMM to multi-layered media case. We hope that the basic framework established here will be useful in those contexts as well.

REFERENCES

[1] S. Tussupbayev, N. Govind, K. Lopata, and C. J. Cramer, "Comparison of real-time and linear-response time-dependent density functional theories for molecular chromophores ranging from sparse to high densities of states," *Journal of Chemical Theory and Computation*, vol. 11, no. 3, pp. 1102–1109, 2015.

[2] M. R. Provorse and C. M. Isborn, "Electron dynamics with real-time time-dependent density functional theory," *International Journal of Quantum Chemistry*, 2016.

[3] F. Wang, C. Y. Yam, L. Hu, and G. Chen, "Time-dependent density functional theory based ehrenfest dynamics," *The Journal of chemical physics*, vol. 135, no. 4, p. 044126, 2011.

[4] G. Bao and J. Lai, "Radar cross section reduction of a cavity in the ground plane," *Communications in Computational Physics*, vol. 15, no. 04, pp. 895–910, 2014.

[5] W. Parnell, I. Abrahams, and P. Brazier-Smith, "Effective properties of a composite half-space: Exploring the relationship between homogenization and multiple-scattering theories.," *Quarterly Journal of Mechanics & Applied Mathematics*, vol. 63, no. 2, 2010.

[6] Y. Wu and Z.-Q. Zhang, "Dispersion relations and their symmetry properties of electromagnetic and elastic metamaterials in two dimensions," *Physical Review B*, vol. 79, no. 19, p. 195111, 2009.

[7] D. J. Griffiths, *Introduction to electrodynamics*. Prentice Hall, 1962.

[8] E. M. Purcell and D. J. Morin, *Electricity and magnetism*. Cambridge University Press, 2013.

[9] J. D. Jackson, *Classical electrodynamics*. John Wiley & Sons, 2007.

[10] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman Lectures on Physics, Desktop Edition Volume I*, vol. 1. Basic books, 2013.

[11] U. Kaldor and S. Wilson, *Theoretical chemistry and physics of heavy and superheavy elements*, vol. 11. Springer Science & Business Media, 2013.

[12] A. Szabo and N. S. Ostlund, *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.

[13] M. A. Marques, N. T. Maitra, F. M. Nogueira, E. K. Gross, and A. Rubio, *Fundamentals of time-dependent density functional theory*, vol. 837. Springer Science & Business Media, 2012.

[14] R. Gerber, V. Buch, and M. A. Ratner, "Time-dependent self-consistent field approximation for intramolecular energy transfer. i. formulation and application to dissociation of van der waals molecules," *The Journal of Chemical Physics*, vol. 77, no. 6, pp. 3022–3030, 1982.

[15] J. Tully, "Mixed quantum–classical dynamics," *Faraday Discussions*, vol. 110, pp. 407–419, 1998.

[16] F. A. Bornemann, P. Nettesheim, and C. Schütte, "Quantum-classical molecular dynamics as an approximation to full quantum dynamics," *The Journal of chemical physics*, vol. 105, no. 3, pp. 1074–1083, 1996.

[17] G. Wentzel, "Eine verallgemeinerung der quantenbedingungen für die zwecke der wellenmechanik," *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 38, no. 6, pp. 518–529, 1926.

[18] M. Marques, *Time-dependent density functional theory*, vol. 706. Springer Science & Business Media, 2006.

[19] E. Runge and E. K. Gross, "Density-functional theory for time-dependent systems," *Physical Review Letters*, vol. 52, no. 12, p. 997, 1984.

[20] E. Gross and W. Kohn, "Time-dependent density-functional theory," *Advances in quantum chemistry*, vol. 21, pp. 255–291, 1990.

[21] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, "An efficient implementation of time-dependent density-functional theory for the calculation of excitation energies of large molecules," *The Journal of Chemical Physics*, vol. 109, no. 19, pp. 8218–8224, 1998.

[22] M. Petersilka, U. Gossmann, and E. Gross, "Excitation energies from time-dependent density-functional theory," *Physical Review Letters*, vol. 76, no. 8, p. 1212, 1996.

[23] M. Cossi and V. Barone, "Time-dependent density functional theory for molecules in liquid solutions," *The Journal of chemical physics*, vol. 115, no. 10, pp. 4708–4717, 2001.

[24] K. Burke, J. Werschnik, and E. Gross, "Time-dependent density functional theory: Past, present, and future," *The Journal of Chemical Physics*, vol. 123, no. 6, p. 062206, 2005.

[25] G. Kresse and D. Joubert, "From ultrasoft pseudopotentials to the projector augmented-wave method," *Physical Review B*, vol. 59, no. 3, p. 1758, 1999.

[26] G. Kresse and J. Hafner, "Norm-conserving and ultrasoft pseudopotentials for first-row and transition elements," *Journal of Physics: Condensed Matter*, vol. 6, no. 40, p. 8245, 1994.

[27] J. Ren, N. Vukmirović, and L.-W. Wang, "Nonadiabatic molecular dynamics simulation for carrier transport in a pentathiophene butyric acid monolayer," *Physical Review B*, vol. 87, no. 20, p. 205117, 2013.

[28] A. Schleife, E. W. Draeger, Y. Kanai, and A. A. Correa, "Plane-wave pseudopotential implementation of explicit integrators for time-dependent kohn-sham equations in large-scale simulations," *The Journal of chemical physics*, vol. 137, no. 22, p. 22A546, 2012.

[29] E. Faou, L. Gauckler, and C. Lubich, "Plane wave stability of the split-step fourier method for the nonlinear schrödinger equation," in *Forum of Mathematics, Sigma*, vol. 2, p. e5, Cambridge Univ Press, 2014.

[30] M. Thalhammer, "High-order exponential operator splitting methods for time-dependent schrödinger equations," *SIAM Journal on Numerical Analysis*, vol. 46, no. 4, pp. 2022–2038, 2008.

[31] D. Gottlieb and S. A. Orszag, *Numerical analysis of spectral methods: theory and applications*. SIAM, 1977.

[32] J.-Y. Lee and L. Greengard, "A fast adaptive numerical method for stiff two-point boundary value problems," *SIAM Journal on Scientific Computing*, vol. 18, no. 2, pp. 403–429, 1997.

[33] L. Greengard, "Spectral integration and two-point boundary value problems," *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 1071–1080, 1991.

[34] L. N. Trefethen, *Spectral methods in MATLAB*. SIAM, 2000.

[35] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, vol. 31. Springer Science & Business Media, 2006.

[36] E. Hairer, "Backward analysis of numerical integrators and symplectic methods," *Annals of Numerical Mathematics*, vol. 1, pp. 107–132, 1994.

[37] E. Hairer and C. Lubich, "Long-time energy conservation of numerical methods for oscillatory differential equations," *SIAM journal on numerical analysis*, vol. 38, no. 2, pp. 414–441, 2000.

[38] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics*, vol. 40, no. 2, pp. 241–266, 2000.

[39] W. Auzinger, H. Hofstätter, W. Kreuzer, and E. Weinmüller, "Modified defect correction algorithms for odes. part i: General theory," *Numerical Algorithms*, vol. 36, no. 2, pp. 135–155, 2004.

[40] A. Christlieb, B. Ong, and J.-M. Qiu, "Integral deferred correction methods constructed with high order runge-kutta integrators," *Mathematics of Computation*, vol. 79, no. 270, pp. 761–783, 2010.

[41] M. Emmett and M. Minion, "Toward an efficient parallel in time method for partial differential equations," *Communications in Applied Mathematics and Computational Science*, vol. 7, no. 1, pp. 105–132, 2012.

[42] D. Beylkin, *Spectral Deferred Corrections for Parabolic Partial Differential Equations*. PhD thesis, YALE UNIVERSITY, 2015.

[43] J. Huang, J. Jia, and M. Minion, "Accelerating the convergence of spectral deferred correction methods," *Journal of Computational Physics*, vol. 214, no. 2, pp. 633–656, 2006.

[44] J. Jia and J. Huang, "Krylov deferred correction accelerated method of lines transpose for parabolic problems," *Journal of Computational Physics*, vol. 227, no. 3, pp. 1739–1753, 2008.

[45] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[46] I. Harari and E. Turkel, "Accurate finite difference methods for time-harmonic wave propagation," *Journal of Computational Physics*, vol. 119, no. 2, pp. 252–270, 1995.

[47] F. Ihlenburg, *Finite element analysis of acoustic scattering*, vol. 132. Springer Science & Business Media, 2006.

[48] P. K. Kythe, *An introduction to boundary element methods*, vol. 4. CRC press, 1995.

[49] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of computational physics*, vol. 114, no. 2, pp. 185–200, 1994.

[50] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations," *IEEE transactions on Electromagnetic Compatibility*, no. 4, pp. 377–382, 1981.

[51] T. Huttunen, M. Malinen, J. P. Kaipio, P. J. White, and K. Hynynen, "A full-wave helmholtz model for continuous-wave ultrasound transmission," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 52, no. 3, pp. 397–409, 2005.

[52] L. Greengard and M. Moura, "On the numerical evaluation of electrostatic fields in composite materials," *Acta numerica*, vol. 3, pp. 379–410, 1994.

[53] R. B. Guenther and J. W. Lee, *Partial differential equations of mathematical physics and integral equations*. Courier Corporation, 1996.

[54] Y. Zhou, M. Feig, and G.-W. Wei, "Highly accurate biomolecular electrostatics in continuum dielectric environments," *Journal of Computational Chemistry*, vol. 29, no. 1, pp. 87–97, 2008.

[55] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[56] Y. Saad, "A flexible inner-outer preconditioned gmres algorithm," *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 461–469, 1993.

[57] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of computational physics*, vol. 73, no. 2, pp. 325–348, 1987.

[58] M. OâNeil, L. Greengard, and A. Pataki, "On the efficient representation of the half-space impedance greenâs function for the helmholtz equation," *Wave Motion*, vol. 51, no. 1, pp. 1–13, 2014.

[59] J. Lai, M. Kobayashi, and L. Greengard, "A fast solver for multi-particle scattering in a layered medium," *Optics express*, vol. 22, no. 17, pp. 20481–20499, 2014.

[60] J. Lai, L. Greengard, and M. O'Neil, "A new hybrid integral representation for frequency domain scattering in layered media," *Applied and Computational Harmonic Analysis*, 2016.

[61] W. Cai and T. Yu, "Fast calculations of dyadic green's functions for electromagnetic scattering in a multilayered medium," *Journal of computational Physics*, vol. 165, no. 1, pp. 1–21, 2000.

[62] S. Chandler-Wilde, "The impedance boundary value problem for the helmholtz equation in a half-plane," *Mathematical Methods in the Applied Sciences*, vol. 20, no. 10, pp. 813–840, 1997.

[63] S. Chandler-Wilde and D. Hothersall, "Efficient calculation of the green function for acoustic propagation above a homogeneous impedance plane," *Journal of Sound and Vibration*, vol. 180, no. 5, pp. 705–724, 1995.

[64] D. Colton and R. Kress, *Integral equation methods in scattering theory*. SIAM, 2013.

[65] M. Durán, R. Hein, and J.-C. Nédélec, "Computing numerically the greenâs function of the half-plane helmholtz operator with impedance boundary conditions," *Numerische Mathematik*, vol. 107, no. 2, pp. 295–314, 2007.

[66] K. Sarabandi and I.-S. Koh, "Fast multipole representation of green's function for an impedance half-space," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 1, pp. 296–301, 2004.

[67] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, vol. 61. Siam, 1998.

[68] D. A. Micha, "An introduction to numerical analysis (atkinson, kendall e.," *J. Chem. Educ*, vol. 57, no. 4, p. A142, 1980.

[69] G. Wanner and E. Hairer, *Solving ordinary differential equations II*, vol. 1. Springer-Verlag, Berlin, 1991.

[70] C. T. Kelley, *Solving nonlinear equations with Newton's method*, vol. 1. Siam, 2003.

[71] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, "Using krylov methods in the solution of large-scale differential-algebraic systems," *SIAM Journal on Scientific Computing*, vol. 15, no. 6, pp. 1467–1488, 1994.

[72] S. Li and L. Petzold, "Software and algorithms for sensitivity analysis of large-scale differential algebraic systems," *Journal of computational and applied mathematics*, vol. 125, no. 1, pp. 131–145, 2000.

[73] E. Hairer, C. Lubich, and M. Roche, "The numerical solution of differential-algebraic systems by runge-kutta methods," 1989.

[74] W. M. Lioen, J. J. de Swart, and W. A. van der Veen, "Test set for ivp solvers," *Report-Department of Numerical Mathematics*, no. 15, pp. 3–1, 1996.

[75] J. M. Hyman and B. Nicolaenko, "The kuramoto-sivashinsky equation: a bridge between pde's and dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 18, no. 1-3, pp. 113–126, 1986.

[76] B. Licolaenko, B. Scheurer, and R. Temam, "Some global dynamical properties of the kuramoto-sivashinsky equation: Nonlinear stability and attractors [j]," *Physica D*, vol. 16, no. 3, pp. 155–183, 1985.

[77] A. Schleife, E. W. Draeger, V. M. Anisimov, A. A. Correa, and Y. Kanai, "Quantum dynamics simulation of electrons in materials on high-performance computers," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 54–60, 2014.

[78] A. J. Christlieb, Y. Liu, and Z. Xu, "High order operator splitting methods based on an integral deferred correction framework," *Journal of Computational Physics*, vol. 294, pp. 224–242, 2015.

[79] R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, and R. Krause, "A multi-level spectral deferred correction method," *BIT Numerical Mathematics*, vol. 55, no. 3, pp. 843–867, 2015.

[80] J. Lions, Y. Maday, and G. Turinici, "A"parareal"in time discretization of pde's," *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, vol. 332, no. 7, pp. 661–668, 2001.

[81] M. J. Gander and S. Vandewalle, "Analysis of the parareal time-parallel time-integration method," *SIAM Journal on Scientific Computing*, vol. 29, no. 2, pp. 556–578, 2007.

[82] C. Farhat and M. Chandesris, "Time-decomposed parallel time-integrators: Theory and feasibility studies for uid, structure, and fluid–structure applications," *Int. J. Numer. Meth. Engng*, vol. 58, pp. 1397–1434, 2003.

[83] M. L. Minion, S. A. Williams, T. E. Simos, G. Psihoyios, and C. Tsitouras, "Parareal and spectral deferred corrections," in *AIP Conference Proceedings*, vol. 1048, pp. 388–391, AIP, 2008.

[84] W. Qu, N. Brandon, D. Chen, J. Huang, and T. Kress, "A numerical framework for integrating deferred correction methods to solve high order collocation formulations of odes," *Journal of Scientific Computing*, pp. 1–37, 2015.

[85] D. Chen, J. Huang, and J. Lu, "Integral equation method preconditioned iterative techniques for the cubic schrödinger equation,"

[86] D. Chen, "A new diagonal parallel preconditioner in time,"

[87] V. Pereyna, "Iterated deferred corrections for nonlinear boundary value problems," *Numerische Mathematik*, vol. 11, no. 2, pp. 111–125, 1968.

[88] P. Zadunaisky, "A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations," in *Symposium-International Astronomical Union*, vol. 25, pp. 281–287, Cambridge Univ Press, 1966.

[89] P. E. Zadunaisky, "On the estimation of errors propagated in the numerical integration of ordinary differential equations," *Numerische Mathematik*, vol. 27, no. 1, pp. 21–39, 1976.

[90] J. Huang, J. Jia, and M. Minion, "Arbitrary order krylov deferred correction methods for differential algebraic equations," *Journal of Computational Physics*, vol. 221, no. 2, pp. 739–760, 2007.

[91] D. J. Higham and L. N. Trefethen, "Stiffness of odes," *BIT Numerical Mathematics*, vol. 33, no. 2, pp. 285–303, 1993.

[92] A. Iserles, "On the numerical quadrature of highly-oscillating integrals i: Fourier transforms," *IMA Journal of Numerical Analysis*, vol. 24, no. 3, pp. 365–391, 2004.

[93] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff pdes," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.

[94] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, "Kssolvâa matlab toolbox for solving the kohn-sham equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 2, p. 10, 2009.

[95] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the laplace equation in three dimensions," *Acta numerica*, vol. 6, pp. 229–269, 1997.

[96] L. Ying, G. Biros, and D. Zorin, "A kernel-independent adaptive fast multipole algorithm in two and three dimensions," *Journal of Computational Physics*, vol. 196, no. 2, pp. 591–626, 2004.

[97] D. A. Luke, *Multilevel modeling*, vol. 143. Sage, 2004.

[98] S. W. Raudenbush and A. S. Bryk, *Hierarchical linear models: Applications and data analysis methods*, vol. 1. Sage, 2002.

[99] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[100] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 253–256, IEEE, 2010.

[101] J. Bruna, S. Chintala, Y. LeCun, S. Piantino, A. Szlam, and M. Tygert, "A theoretical argument for complex-valued convolutional networks," *arXiv preprint arXiv:1503.03438*, 2015.

[102] R. D. Blumofe, C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, and Y. Zhou, *Cilk: An efficient multithreaded runtime system*, vol. 30. ACM, 1995.

[103] R. D. Blumofe and C. E. Leiserson, "Scheduling multithreaded computations by work stealing," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 720–748, 1999.

[104] M. Frigo, C. E. Leiserson, and K. H. Randall, "The implementation of the cilk-5 multithreaded language," in *ACM Sigplan Notices*, vol. 33, pp. 212–223, ACM, 1998.

[105] D. Chen, J. Huang, H. Wang, X. Yue, and B. Zhang, "A parallel adaptive recursive solver for ode two-point boundary value problems,"

[106] M. H. Cho, J. Huang, D. Chen, and W. Cai, "A heterogeneous fmm for 2-d layered media helmholtz equation i: Two & three layers cases," *arXiv preprint arXiv:1703.09136*, 2017.

[107] R. Courant and D. Hilbert, *Methods of mathematical physics*, vol. 1. CUP Archive, 1965.

[108] W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, "Remarks on the implementation of wideband fmm for the helmholtz equation in two dimensions," *Contemporary Mathematics*, vol. 408, pp. 99–110, 2006.

[109] K. L. Ho, *Fast direct methods for molecular electrostatics*. PhD thesis, New York University, 2012.

[110] L. Greengard and J. Strain, "A fast algorithm for the evaluation of heat potentials," *Communications on Pure and Applied Mathematics*, vol. 43, no. 8, pp. 949–963, 1990.

[111] J.-R. Li and L. Greengard, "High order accurate methods for the evaluation of layer heat potentials," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3847–3860, 2009.

[112] W. C. Chew, *Waves and fields in inhomogeneous media*, vol. 522. IEEE press New York, 1995.

[113] M. H. Cho and W. Cai, "Efficient and accurate computation of electric field dyadic greenâs function in layered media," *Journal of Scientific Computing*, vol. 71, no. 3, pp. 1319–1350, 2017.

[114] T. J. Cui and W. C. Chew, "Fast evaluation of sommerfeld integrals for em scattering and radiation by three-dimensional buried objects," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 887–900, 1999.

[115] K. A. Michalski and D. Zheng, "Electromagnetic scattering and radiation by surfaces of arbitrary shape in layered media. i. theory," *IEEE Transactions on Antennas and propagation*, vol. 38, no. 3, pp. 335–344, 1990.